[EAMT May 2010 St Raphael, France]

# Chunk-Based EBMT

**Jae Dong Kim** **Ralf D. Brown** **Jaime G. Carbonell**

**Language Technologies Institute**
Carnegie Mellon University
Pittsburgh, PA 15213
{jdkim,ralf,jgc}@cs.cmu.edu

## Abstract

Corpus driven machine translation approaches such as Phrase-Based Statistical Machine Translation and Example-Based Machine Translation have been successful by using word alignment to find translation fragments for matched source parts in a bilingual training corpus. However, they still cannot properly deal with systematic translation for insertion or deletion words between two distant languages. In this work, we used syntactic chunks as translation units to alleviate this problem, improve alignments and show improvement in BLEU for Korean to English and Chinese to English translation tasks.

## 1 Introduction

In general, corpus-based machine translation systems prefer longer units because they naturally convey local context and local reordering. This was achieved by phrases in Phrase-Based Statistical Machine Translation (Koehn et al., 2007; Vogel et al., 2003) and surface form matches in lexical Example-Based Machine Translation (Brown, 2005; Veale and Way, 1997). These systems use phrasal alignment to find translations of matched n-grams for an input sentence.

However, because the alignment algorithms used in these systems purely depend on word alignment (Brown et al., 1993) they cannot address structural translations, other than hoping for structural parallelism between source and target. For example, these algorithms cannot reliably find 'an/the office' as a translation of 'sa-moo-sil' in Korean to English translation because Korean does not have articles.

For this reason, we investigated the chunk as our translation unit. The chunk was pioneered by Abney (1991). It is a continuous and non-recursive syntactic segment around a head, comparable to a morphologically complex word in synthetic languages, and is not contained in any other chunks. A typical chunk consists of a single content word surrounded by related function words matching a template, though it could contain modifiers.

We have observed several advantages by using chunks as basic translation units. First, we can to some degree systematically translate untranslatable tokens (words, morphemes) that exist only in one side. For example, when we translate an English sentence into Korean, the word-to-word translation systems cannot produce a nominative case marker in Korean unless rules are given by human experts or the MT system "hallucinates" markers and uses the target language modeling to guess whether or not the case marker should in fact be present. Second, as chunks are n-gram phrases, they convey local reordering and context as well, though this advantage is also true of n-grams in phrasal translation. Third, the number of chunks may better match across languages than the number of words, which may yield better alignment at the chunk level. Fourth, since the order of chunks is more flexible than the order of words within a chunk, it 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 relatively less strict word-order language than English or the Romance languages.

In this paper, we show that we obtain significant improvements using chunks for translation in both

Korean to English and Chinese to English.

We discuss related work in section 2, our approaches for chunk alignment and translation in section 3. In section 4, we describe our experiments, and we discuss our conclusions in section 5.

## 2 Related Work

Some researchers have studied exploiting chunks in translation. Le et al. (2000) used chunk alignment to get better word alignment. Given a human dictionary and chunked English sentences, they got corresponding Chinese sentences chunked via chunk projection.

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

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

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

Wu (1997) studied the 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 to extract 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 decomposed 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 decomposed $P(J, \mathcal{J}, \mathcal{E}|E)$ 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) decomposed the translation model into sentence level reordering (SLR), chunk mapping (CM) and word translations (W):

$$\begin{aligned} p(f|e) &= p(SLR|e) \times \prod_i p(CM_i|e, SLR) \\ &\quad \times \prod_j (W_{ij}|CM_i, SLR, e) \end{aligned} \tag{1}$$

Sentence level chunk reordering defines how source and target chunks are connected and chunk mapping defines an alignment of source to target POSs. Finally word translations set the lexical composition of the target language sentence. They reported improved performance over IBM Model 4 on a short sentence translation task.

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

First, we use existing monolingual chunkers which use machine learning techniques to find chunk boundaries and thus are driven by a hand annotated training corpus. Most automated chunk detection algorithms in MT are bilingual and heavily depend on human resources such as human dictionaries (Le et al., 2000; Hwang et al., 2004) and hand-written grammars (Wu, 1997) and others depend on co-occurrence statistics either bilingually or mono-lingually (Zhou et al., 2004; Watanabe et al., 2003). These approaches can be less accurate when resources are limited. In this work, we use existing chunkers to avoid errors that can be caused by insufficient resources, inaccurate alignments and lack of a gold standard in training. However, since we use monolingual chunkers, we do not maximize chunk correspondence between source and target languages. It is possible that a hybrid approach which also maximizes cross-lingual chunk correspondence would be even better than our method.

Second, we developed a new chunk alignment algorithm that is tightly combined with IBM word alignment models. The basic idea is to apply well-known IBM word alignment algorithms to chunk alignment by treating a chunk as a word and boosting chunk alignment with word alignment. Since a chunk typically consists of multiple words and occurs with lower frequency than individual words, we would normally need a huge parallel corpus in which we can find reasonable statistical evidence to align chunks. However, our approach allows a relatively smaller corpus by boosting chunk alignment with word alignment information, making it practical for low and medium resource conditions.

Last, in decoding, our method combines target chunks as well as target fragments which are not chunks. Unlike the lexical EBMT system by Brown (2005), this chunk-based system is a hybrid system that combines a typical string-based EBMT system and a chunk-based EBMT system. It is close to the EBMT system by Veale and Way (1997) and Stroppa and Way (2006) in that it uses constituent-like units but different in that the work is fully extended to include chunks at all levels. Our work also differs from the ChunkMT work (Koehn and Knight, 2002), in which the translation was decomposed into sentence label chunk reordering, chunk mapping and word translation. When an input is given, they build a sentence level template first and then use chunk mapping and word translation to generate a target translation. Whereas the method is good, each step could be prone to error, and errors could compound. Our system relies on chunks as the basic unit when it can find evidence of good chunk level translations, but otherwise it falls back to a word/phrase-based model.

## 3 Chunk-Based System

### 3.1 Chunk Detection

In this work, we used in-house monolingual chunkers rather than using synchronous chunkers. We built the chunking models using Conditional Random Fields (Lafferty et al., 2001) giving hand chunked sentences for Korean and Chinese. For English, we used a SNoW based shallow parser (Carlson et al., 1999) for chunking.

### 3.2 Chunk Alignment

In general, aligning chunks is a harder task than aligning words if we use an unsupervised method such as IBM Model 4, when training data is limited. The reason is that we have many more word tokens than chunk tokens in a corpus and thus less statistical evidence for chunk alignment. For example, 'in the office' is a chunk and appears much less often than each of the comprising words 'in', 'the' and 'office' in a corpus. So statistical evidence for alignment for the chunk is less obvious than that of each comprising word and results in poorer alignments. Hence aligning words and deriving chunk alignment using this alignment information is a useful process unless we have a sufficiently large corpus for chunk alignment.

But in reality, it is hard to build such a large corpus for many languages. Instead, we investigated a new method that induces chunk alignment from word alignment.

For chunk alignment, we used GIZA++ (Och and Ney, 2003) that works on a chunk annotated corpus. We created a chunk unit which will be used as a basic unit in GIZA++ by concatenating all the words in the chunk placing a special delimiter character between adjacent words.

The GIZA++ was modified to use word alignment information in chunk alignment in this way:

- 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

$$T(\mathbf{f}|\mathbf{e}) = \frac{C(\mathbf{f},\mathbf{e})}{\sum_k C(\mathbf{f}_k,\mathbf{e})} \quad (2)$$

- We redefine it as,

$$T(\mathbf{f}|\mathbf{e}) = \frac{C'(\mathbf{f},\mathbf{e})}{\sum_k C'(\mathbf{f}_k,\mathbf{e})} \quad (3)$$

where

$$C'(\mathbf{f},\mathbf{e}) = C(\mathbf{f},\mathbf{e}) \times F(\mathbf{f},\mathbf{e}) \quad (4)$$

where $F(\mathbf{f},\mathbf{e})$ is a weighting function and for this, we used power means with powers $\geq 2$ in this work:

$$F(\mathbf{f},\mathbf{e}) = \left( \frac{1}{m} \sum_{j=1}^{m} max_{i=1}^{n} (T(f_i|e_j))^p \right)^{\frac{1}{p}} \quad (5)$$

.

### 3.3 Translation by Chunks

In our work, the source and target chunks we used are not detected synchronously. Therefore, we have a large number of one-to-many, many-to-one and many-to-many relationships between source chunks and target chunks. So we find consistent chunk sequence pairs as translation pairs using the *Refined Method* that Och and Ney (2003) used for phrase extraction. We explain this with the version implemented by Koehn (2004). We start with the intersection of the two chunk alignments, adding new alignment points that exist in the union of two chunk alignments and connect at least one previously unaligned chunk. First, we expand to directly adjacent alignment points. We check for potential points starting from the bottom left corner of the alignment matrix, checking for alignment points for the first target chunk, then continue 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, with otherwise the same requirements. We collect all aligned chunk sequence pairs that are consistent with the chunk alignment: The chunks in a legal chunk sequence pair are only aligned to each other, and not to chunks outside.

Figure 1 illustrates how the *Refined Method* refined chunk alignment and how chunk translation sequence pairs are extracted afterwards on a Korean and English sentence pair. The transliteration of the Korean sentence is “[jeo] [,] [aek-jeong-pae-neol] [joo-moon e] [gwan-hae] [jeon-hwa-deu-ryeot-neun-dae-yo] [.]” which literally means “[well] [,] [lcd] [order to] [related/about] [am calling] [.]”. [1] 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*. After alignment refinement is done, chunk translation sequence pairs are extracted based on the alignment. The rectangular areas denote extracted chunk sequence translation pairs.

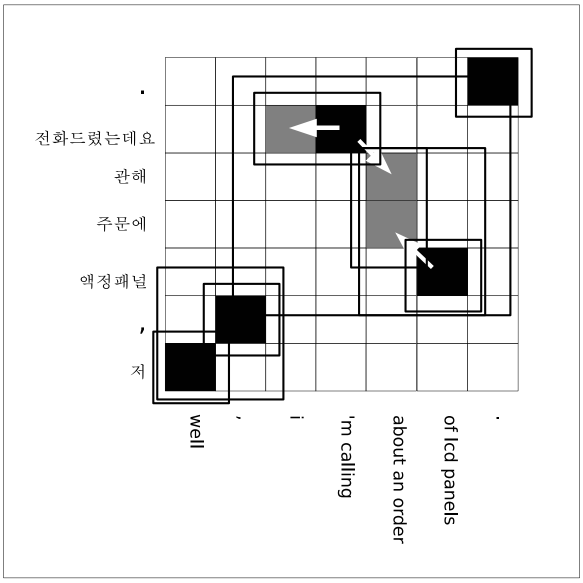


Figure 1: Chunk Translation Sequence Pair Extraction

| Lang. | Data | Moses | EBMT |
|---|---|---|---|
| Kr-En | Dev | 0.2222 | **0.2382** |
| | Unseen | 0.2289 | **0.2502** |
| Cn-En | Dev | **0.2610** | 0.2538 |
| | Unseen | **0.2533** | 0.2295 |

Table 1: Baseline System vs. Moses

## 3.4 System Integration

### 3.4.1 Baseline System

Our baseline system is a lexical EBMT system that uses surface form match as a similarity function. Given an input sentence to translate, the system first performs surface form matching over the source half of the training set. Then it finds their translations using a phrasal aligner, and combines them to form hypothesis translations. The aligner provides several feature values to a standard beam decoder. [2] Next, 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 hypothesis translation. The score is calculated as a combination of feature values with their weights tuned in a separate tuning process in a log linear model.

The performance of this EBMT system is comparable to that of Moses system (Koehn et al., 2007) given the same data. Table 1 shows the translation performance on the development sets. (The data sets are described in section 4.1.) In Korean-English translation, EBMT performs better while Moses is better in Chinese-English translation.

### 3.4.2 Chunk-Based System

Figure 2 shows how the components are integrated to build a Chunk-Based EBMT system. And the system works in this way: Firstly, given an input sentence, the system finds chunk sequence matches and a chunk aligner finds their translations from the stored consistent chunk sequence pairs and calculates feature values for them as the phrasal aligner in the baseline system does. Sec-

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

[2] The aligner's feature scores include uni-directional SPA scores (Kim et al., 2005), number of untranslated words on both source and target sides and so on.

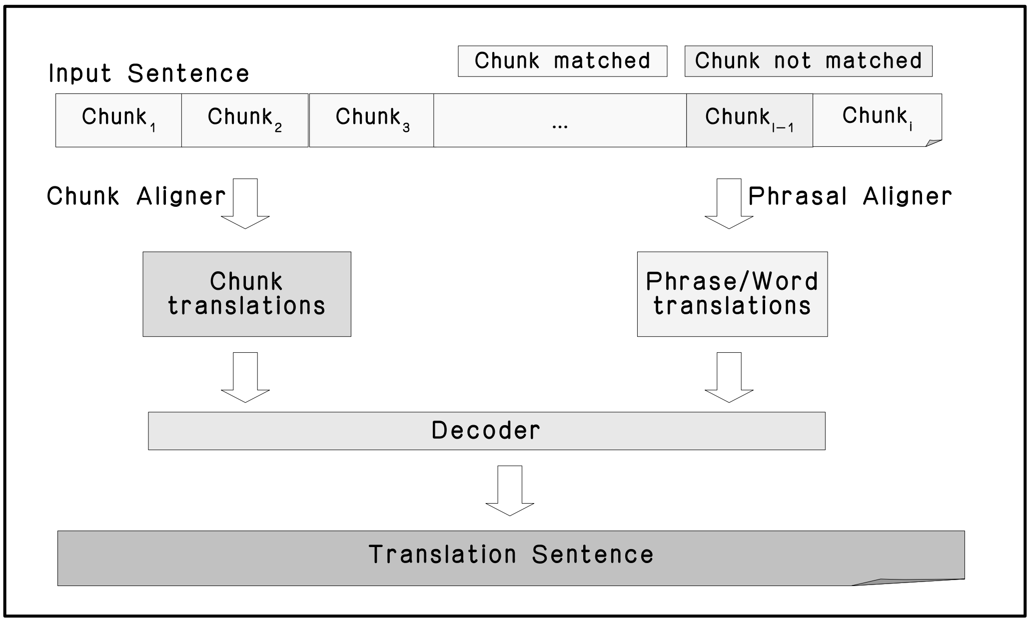


Figure 2: System Integration

ondly, when no chunk match is found or no chunk alignment is found, it behaves as the baseline system (i.e., it finds word/phrase matches and use a phrasal aligner to find the translations of them.) Thirdly, it puts chunk translations and word/phrase translations into a lattice. For each translation, it adds an additional feature that indicates whether the translation is from the chunk alignment or not. Finally, it performs standard beam decoding to find the best translation.

## 4 Evaluation

The chunk-based approach can be more beneficial for a distant language pair. When we have a very similar language pair in terms of sentence structures and word correspondence, we have very accurate alignment and this leads to high quality translation. However, when we have a very distant language pair, it is much harder to align words due to lower sentence structure agreement and word correspondence. Consequently the translation quality is much lower. This problem is alleviated by chunks because a sentence pair has a higher correspondence at the chunk level. So if we align chunks in a distant language pair and use them in translation, we can have better translation quality.

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

Korean is a SOV language where a verb follows an accusative while English is classified as a SVO language. It has many case markers that do not exist in English, but it lacks some of the English functional words such as articles. For this reason, 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 'a' or a 'the' in front of 'office' depending on the context. Also, when there is a correspondent for a case marker, their positions are different which leads to reordering difficulty. In English, a preposition comes before a head word but its correspondent case marker in Korean follows the head word.

Although Chinese is classified as an SVO language like English, it is very different from English in that it is a topic-prominent language, it has aspect and mood particles, and it requires a classifier in counting nouns. It also lacks many correspondents of English function words. So if we translate Chinese to English by chunks, we are likely to have the benefit of translating inserted words naturally.

We compare the chunk-based system with a lexical EBMT system in this evaluation. The lexical EBMT system uses a phrasal alignment algorithm for arbitrary matches and serves as a baseline system in these experiments.

We use BLEU (Papineni et al., 2002) as our

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

Table 2: Training set for Korean-English

| | sentences | words | chunks | #ref |
|---|---|---|---|---|
| Dev | 966 | 6,071 | 8,591 | 1 |
| Unseen | 1,170 | 7,422 | 10,441 | 1 |

Table 3: Test sets for Korean-English

translation evaluation metric.

### 4.1 Data

For Korean-English, we used 28,000 sentence pairs as a training set and 966 sentences with 1 reference translation for parameter tuning. As an unseen set, we used 1170 sentences with 1 reference translation.

The data consists of conversational sentences from the travel and business domains. So the sentences are very short: the Korean sentences average 6.5 chunks and 8.9 words long. and the English sentences average 6.4 chunks and 9.5 words long. The chunks are 1.4 words long and 1.5 words long on average in Korean and English respectively. Table 2 describes the training data we used in this experiment.

Table 3 shows the test sets we used for parameter tuning (Dev) and unseen data performance (Unseen). These are in-domain data sets and thus are very similar in sentence and chunk lengths. The Dev set has 6.3 chunks and 8.9 words per sentence, and its chunks are 1.4 words long on average. The Unseen set also has 6.3 chunks and 8.9 words long sentences and its chunks are 1.4 words long on average. [3]

Table 4 shows the coverage of the training set on the test sets in Korean. We calculated word, chunk and multi-word chunk coverages. First, the word coverage is to see what portion of the input can be translated in a normal word/phrase based translation system. Second, the chunk coverage is to see how big a portion can be translated by chunks. Last, we measured the multi-word chunk coverage because we are mostly likely to have benefits by translating chunks which are longer than one word by properly dealing with word deletion/insertion as explained.

[3]Korean words were counted after splitting morphemes found by MACH (Shim and Yang, 2002) Korean morphological analyzer. However we did not split morphemes in Korean verbs and adjectives because they are morphologically very complicated.

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

Table 4: Training Set Coverage for Korean

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

Table 5: Training set for Chinese-English

For Chinese-English translation, we used 340,000 sentence pairs for training and 230 sentences with 4 references for parameter tuning. To measure performance on an unseen data set, we used 919 sentences with 4 references from NIST Machine Translation Evaluation 2003 (NIST, 2003).

The Chinese training data was drawn from the FBIS Chinese-English parallel text by (NIST, 2003). We used sentence pairs with 70 or fewer words in the source side. On average, the Chinese sentences are 26.8 words long and 18.1 chunks long and chunks are composed of 1.5 words. And the English sentences are 33.9 words long and 18 chunks long and chunks are 1.8 words long. Table 5 shows Chinese to English training data. Both Korean-English and Chinese-English language pairs shows chunk level correspondence is higher than word level correspondence.

Table 6 shows Chinese-English test sets for translation parameter tuning (Dev) and translation performance evaluation on unseen data (Unseen). The training set coverages look similar to those of Korean.

### 4.2 Results and Analysis

Table 7 shows performance comparisons between the baseline system (EBMT) and the new Chunk-Based EBMT (EBMT(C)). For both the development set and unseen set, EBMT(C) performs better than the baseline system in Korean to English translation. The Chunk-based system also per-

| | sentences | words | chunks | #ref |
|---|---|---|---|---|
| Dev | 230 | 4,076 | 6,089 | 4 |
| Unseen | 919 | 16,083 | 24,106 | 4 |

Table 6: Test sets for Chinese-English

| Lang. | Data | EBMT | EBMT(C) |
|---|---|---|---|
| Kr-En | Dev | 0.2382 | **0.2452** |
| | Unseen | 0.2502 | **0.2522** |
| Cn-En | Dev | 0.2538 | **0.2631** |
| | Unseen | 0.2295 | **0.2427** |

Table 7: Chunk-Based EBMT Translation Performance (BLEU)

forms better than the baseline system in Chinese to English translation. The BLEU scores in bold means they are significantly better.

We then analyzed the effect of chunk in translation. In Korean-English translation, we took subsets out of the Dev set and its translation in this way: For each source sentence and translation pair $(s_i, t_i)$, $C(i)$ counts the case when there is a chunk match and the target chunk translation appears in the reference. And $P(i)$ counts the case when the target chunk translation is also found by the phrasal aligner in the baseline system. Finally we extracted a subset of translations based on the difference $C(i) - P(i)$, and calculated the performance difference between the baseline system and the new system for the set.

Table 8 shows our observation. From the whole hypothesis translation set, we picked the maximal subset of translations that satisfies the criterion in each step and calculated a BLEU score. As shown in the table, as the difference $C(i)-P(i)$ increases, the BLEU score difference increases as well. This obviously shows that the aligned chunk pairs may be helpful for translation, as they improve BLEU scores.

## 5 Discussion and Future Work

In this paper, we showed that our chunk based translation system outperforms the baseline lexical EBMT system. We developed a chunk alignment algorithm that combines evidence for chunk alignment with evidence for constituent word alignment to boost performance especially for low-resource MT tasks. From the chunk alignment information, we recognize consistent chunk sequence pairs as translation pairs. These chunk translation pairs were used together with phrasal alignment in translation and helped the translation system systematically.

Although the chunk alignment algorithm was implicitly evaluated in its positive contributions to translation, we did not present a direct evaluation of alignment, due to lack of a meaningful gold standard. But, a direct evaluation for the chunk alignment algorithm may be of interest especially if used beyond MT.

In this work, we designed the MT system to fall back to the lexical/phrasal EBMT when lacking chunk coverage, and when corresponding target chunks do not align with sufficient confidence. But we think it will be interesting and helpful to generate somewhat more generalized chunk-templates as translation pairs from the chunks found in training translations. For example, if we don't have a match for “in the office” but have “in the school” and “office” as already found translations, we can generate the translation of “in the office” from the translations of “in the school” and “office”. This will be more helpful in Chinese-English translation because the chunk coverage by the training set in Chinese is lower.

We used relatively small data in these experiments, since our focus was on improving resource-constrained MT. But considering that chunk alignment alleviates some structural mismatch problems in a language pair, we think the method may still have improvements with a larger data set.

## References

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

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, June.

Brown, Ralf D. 2005. 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. http://www.cs.cmu.edu/~ralf/papers.html.

Carlson, A., C. Cumby, J. Rosen, and D. Roth. 1999. The SNoW learning architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department, May.

| | sentences | EBMT | EBMT(C) | Difference |
|---|---|---|---|---|
| Whole | 966 | 0.2382 | 0.2451 | 0.0069 |
| $\{t_i \| C(i) > 0\}$ | 963 | 0.2386 | 0.2456 | 0.0070 |
| $\{t_i \| C(i) - P(i) >= 1\}$ | 493 | 0.2928 | 0.3038 | 0.0110 |
| $\{t_i \| C(i) - P(i) >= 2\}$ | 202 | 0.3425 | 0.3574 | 0.0149 |
| $\{t_i \| C(i) - P(i) >= 3\}$ | 71 | 0.3651 | 0.3827 | 0.0176 |

Table 8: Effect of Chunk Translation for Korean to English (BLEU)


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

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

Koehn, Philipp and Kevin Knight. 2002. ChunkMT: Statistical machine translation with richer linguistic knowledge.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2007)*, June. demonstration session.

Koehn, Philipp. 2004. 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.

Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Le, S., J. Youbing, D. Lin, and S. Yufang. 2000. Word alignment of english-chinese bilingual corpus based on chunks.

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

NIST. 2003. Machine translation evaluation. `http://nist.gov/speech/tests/mt/`.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51.

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

Shim, Kwangseob and Jaehyung Yang. 2002. Mach: a supersonic korean morphological analyzer. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Stroppa, Nicolas and Andy Way. 2006. Matrex: Dcu machine translation system for iwslt 2006.

Veale, Tony and Andy Way. 1997. 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. `http://www.compapp.dcu.ie/~tonyv/papers/gaijin.html`.

Vogel, Stephan, Ying Zhang, Fei Huang, Alicia Tribble, Ashish Venugopal, Bing Zhao, and Alex Waibel. 2003. The cmu statistical machine translation system. In *Proceedings of the Ninth Machine Translation Summit*, September. `http://www.amtaweb.org/summit/MTSummit/papers.html`.

Watanabe, Taro, Eiichiro Sumita, and Hiroshi G. Okuno. 2003. Chunk-Based statistical translation. In Hinrichs, Erhard and Dan Roth, editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 303–310, Sapporo, Japan.

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

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