Chunk-based Statistical Translation

Taro Watanabe†, Eiichiro Sumita† and **Hiroshi G. Okuno‡**
{taro.watanabe, eiichiro.sumita}@atr.co.jp

† ATR Spoken Language Translation Research Laboratories
2-2-2 Hikaridai, Keihanna Science City
Kyoto 619-0288 JAPAN

‡Department of Intelligence Science and Technology
Graduate School of Informatics, Kyoto Uniersity
Kyoto 606-8501 JAPAN

Abstract

This paper describes an alternative translation model based on a text chunk under the framework of statistical machine translation. The translation model suggested here first performs chunking. Then, each word in a chunk is translated. Finally, translated chunks are reordered. Under this scenario of translation modeling, we have experimented on a broad-coverage Japanese-English traveling corpus and achieved improved performance.

1 Introduction

The framework of statistical machine translation formulates the problem of translating a source sentence in a language J into a target language E as the maximization problem of the conditional probability $\hat{E} = \mathrm{argmax}_E\, P(E|J)$. The application of the Bayes Rule resulted in $\hat{E} = \mathrm{argmax}_E\, P(E)P(J|E)$. The former term $P(E)$ is called a language model, representing the likelihood of E. The latter term $P(J|E)$ is called a translation model, representing the generation probability from E into J.

As an implementation of $P(J|E)$, the word alignment based statistical translation (Brown et al., 1993) has been successfully applied to similar language pairs, such as French–English and German–English, but not to drastically different ones, such as Japanese–English. This failure has been due to the limited representation by word alignment and the weak model structure for handling complicated word correspondence.

This paper provides a chunk-based statistical translation as an alternative to the word alignment based statistical translation. The translation process inside the translation model is structured as follows. A source sentence is first chunked, and then each chunk is translated into target language with local word alignments. Next, translated chunks are reordered to match the target language constraints.

Based on this scenario, the chunk-based statistical translation model is structured with several components and trained by a variation of the EM-algorithm. A translation experiment was carried out with a decoder based on the left-to-right beam search. It was observed that the translation quality improved from 46.5% to 52.1% in BLEU score and from 59.2% to 65.1% in subjective evaluation.

The next section briefly reviews the word alignment based statistical machine translation (Brown et al., 1993). Section 3 discusses an alternative approach, a chunk-based translation model, ranging from its structure to training procedure and decoding algorithm. Then, Section 4 provides experimental results on Japanese-to-English translation in the traveling domain, followed by discussion.

2 Word Alignment Based Statistical Translation

Word alignment based statistical translation represents bilingual correspondence by the notion of word alignment A, allowing one-to-many generation from each source word. Figure 1 illustrates an example of English and Japanese sentences, E and J, with sample word alignments. In this example, "show$_1$" has generated two words, "mise$_5$" and "tekudasai$_6$".

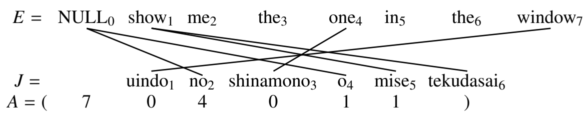

Figure 1: Example of word alignment

Under this word alignment assumption, the translation model $P(J|E)$ can be further decomposed without approximation.

$$P(J|E) = \sum_{A} P(J, A|E)$$

2.1 IBM Model

During the generation process from E to J, $P(J, A|E)$ is assumed to be structured with a couple of processes, such as insertion, deletion and reorder. A scenario for the word alignment based translation model defined by Brown et al. (1993), for instance IBM Model 4, goes as follows (refer to Figure 2).

1. Choose the number of words to generate for each source word according to the Fertility Model. For example, "show" was increased to 2 words, while "me" was deleted.

2. Insert NULLs at appropriate positions by the NULL Generation Model. Two NULLs were inserted after each "show" in Figure 2.

3. Translate word-by-word for each generated word by looking up the Lexicon Model. One of the two "show" words was translated to "mise."

4. Reorder the translated words by referring to the Distortion Model. The word "mise" was reordered to the 5th position, and "uindo" was reordered to the 1st position. Positioning is determined by the previous word's alignment to capture phrasal constraints.

For the meanings of each symbol in each model, refer to Brown et al. (1993).

2.2 Problems of Word Alignment Based Translation Model

The strategy for the word alignment based translation model is to translate each word by generating multiple single words (a bag of words) and to determine the position of each translated word. Although this procedure is sufficient to capture the bilingual correspondence for similar language pairs, some issues remain for drastically different pairs:

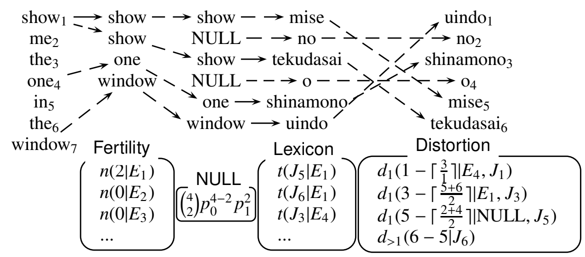

Figure 2: Word alignment based translation model $P(J, A|E)$ (IBM Model 4)

Insertion/Deletion Modeling Although deletion was modeled in the Fertility Model, it merely assigns zero to each deleted word without considering context. Similarly, inserted words are selected by the Lexical Model parameter and inserted at the positions determined by a binomial distribution.

This insertion/deletion scheme contributed to the simplicity of this representation of the translation processes, allowing a sophisticated application to run on an enormous bilingual sentence collection. However, it is apparent that the weak modeling of those phenomena will lead to inferior performance for language pairs such as Japanese and English.

Local Alignment Modeling The IBM Model 4 (and 5) simulates phrasal constraints, although there were implicitly implemented as its Distortion Model parameters. In addition, the entire reordering is determined by a collection of local reorderings insufficient to capture the long-distance phrasal constraints.

The next section introduces an alternative modeling, chunk-based statistical translation, which was intended to resolve the above two issues.

3 Chunk-based Statistical Translation

Chunk-based statistical translation models the process of chunking for both the source and target sentences, E and J,

$$P(J|E) = \sum_{\mathcal{J}} \sum_{\mathcal{E}} P(J, \mathcal{J}, \mathcal{E}|E)$$

where $\mathcal{J}$ and $\mathcal{E}$ are the chunked sentences for J and E, respectively, defined as two-dimentional ar-

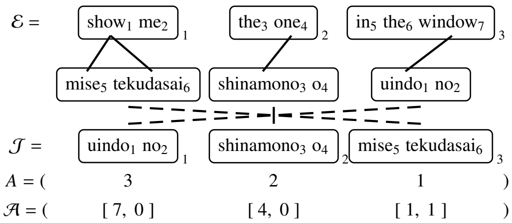

Figure 3: Example of chunk-based alignment

rays. For instance, $\mathcal{J}_{i,j}$ represents the jth word of the ith chunk. The number of chunks for source and target is assumed to be equal, $|\mathcal{J}| = |\mathcal{E}|$, so that each chunk can convey a unit of meaning without added/subtracted information. The term $P(J, \mathcal{J}, \mathcal{E}|E)$ is further decomposed by chunk alignment A and word alignment for each chunk translation $\mathcal{A}$.

$$P(J, \mathcal{J}, \mathcal{E}|E) = \sum_{A} \sum_{\mathcal{A}} P(J, \mathcal{J}, A, \mathcal{A}, \mathcal{E}|E)$$

The notion of alignment A is the same as those found in the word alignment based translation model, which assigns a source chunk index for each target chunk. $\mathcal{A}$ is a two-dimensional array which assigns a source word index for each target word per chunk. For example, Figure 3 shows two-level alignments taken from the example in Figure 1. The target chunk at position 3, $\mathcal{J}_3$, "mise tekudasai" is aligned to the first position ($A_3 = 1$), and both the words "mise" and "tekudasai" are aligned to the first position of the source sentence ($\mathcal{A}_{3,1} = 1$, $\mathcal{A}_{3,2} = 1$).

3.1 Translation Model Structure

The term $P(J, \mathcal{J}, A, \mathcal{A}, \mathcal{E}|E)$ is further decomposed with approximation according to the scenario described below (refer to Figure 4).

1. Perform chunking for source sentence E by $P(\mathcal{E}|E)$. For instance, chunks of "show me" and "the one" were derived. The process is modeled by two steps:
   (a) Selection of chunk size (Head Model). For each word E_i, assign the chunk size φ_i using the head model $\epsilon(\varphi_i|E_i)$. A word with chunk size more than 0 ($\varphi_i > 0$) is treated as a head word, otherwise a non-head (refer to the words in bold in Figure 4).
   (b) Associate each non-head word to a head word (Chunk Model). Each non-head word E_i is associated to a head word E_h by the probability $\eta(c(E_h)|h - i, c(E_i))$, where h is the position of a head word and $c(E)$ is a function to map a word E to its word class (i.e. POS). For instance, "the$_3$" is associated with the head word "one$_4$" located at $4 - 3 = +1$.
2. Select words to be translated with Deletion and Fertility Model.
   (a) Select the number of head words. For each head word E_h ($\varphi_h > 0$), choose fertility ϕ_h according to the Fertility Model $\nu(\phi_h|E_h)$. We assume that the head word must be translated, therefore $\phi_h > 0$. In addition, one of them is selected as a head word at target position using a uniform distribution $1/\phi_h$.
   (b) Delete some non-head words. For each non-head word E_i ($\varphi_i = 0$), delete it according to the Deletion Model $\delta(d_i|c(E_i), c(E_h))$, where E_h is the head word in the same chunk and d_i is 1 if E_i is deleted, otherwise 0.
3. Insert some words. In Figure 4, NULLs were inserted for two chunks. For each chunk $\mathcal{E}_i$, select the number of spurious words ϕ'_i by Insertion Model $\iota(\phi'_i|c(E_h))$, where E_h is the head word of $\mathcal{E}_i$.
4. Translate word-by-word. Each source word E_i, including spurious words, is translated to J_j according to the Lexicon Model, $\tau(J_j|E_i)$.
5. Reorder words. Each word in a chunk is reordered according to the Reorder Model $P(\mathcal{A}_j|\mathcal{E}_{A_j}, \mathcal{J}_j)$. The chunk reordering is taken after the Distortion Model of IBM Model 4, where the position is determined by the relative position from the head word,

$$P(\mathcal{A}_j|\mathcal{E}_{A_j}, \mathcal{J}_j) = \prod_{k=1}^{|\mathcal{A}_j|} \rho(k - h|c(E_{\mathcal{A}_{A_j,k}}), c(\mathcal{J}_{j,k}))$$

where h is the position of a head word for the chunk $\mathcal{J}_j$. For example, "no" is positioned -1 of "uindo".

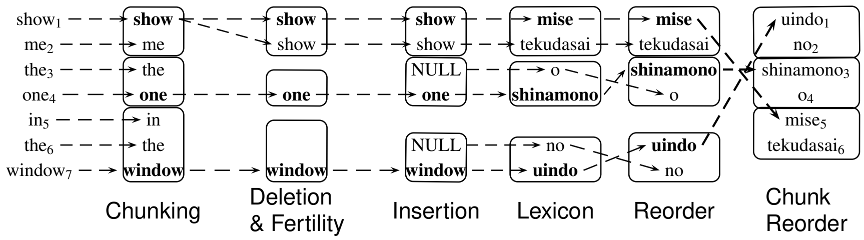

Figure 4: Chunk-based translation model. The words in bold are head words.

6. Reorder chunks. All of the chunks are reordered according to the Chunk Reorder Model, $P(A|\mathcal{E},\mathcal{J})$. The chunk reordering is also similar to the Distortion Model, where the positioning is determined by the relative position from the previous alignment

$$P(A|\mathcal{E},\mathcal{J}) = \prod_{j=1}^{|\mathcal{J}|} \varrho(j - j'|c(\mathcal{E}_{A_{j-1},h'}), c(\mathcal{J}_{j,h}))$$

where j' is the chunk alignment of the the previous chunk a$\mathcal{E}_{A_{j-1}}$. h and h' are the head word indices for $\mathcal{J}_j$ and $\mathcal{E}_{A_{j-1}}$, respectively. Note that the reordering is dependent on head words.

To summarize, the chunk-based translation model can be formulated as

$$\begin{aligned} P(J|E) \quad = \quad & \sum_{\mathcal{E},\mathcal{J},\mathcal{A},A} \prod_i \epsilon(\varphi_i|E_i) \\ & \times \prod_{i:\varphi_i=0} \eta(c(E_{h_i})|h_i - i, c(E_i)) \\ & \times \prod_{i:\varphi_i>0} \nu(\phi_i|E_i)/\phi_i \\ & \times \prod_{i:\varphi_i=0} \delta(d_i|c(E_i), c(E_{h_i})) \\ & \times \prod_{i:\varphi_i>0} \iota(\phi'_i|c(E_i)) \times \prod_j \prod_k \tau(\mathcal{J}_{j,k}|E_{\mathcal{A}_{j,k}}) \\ & \times \prod_j P(\mathcal{A}_j|\mathcal{E}_{A_j},\mathcal{J}_j) \times P(A|\mathcal{E},\mathcal{J}) \end{aligned}$$

.

3.2 Characteristics of chunk-based Translation Model

The main difference to the word alignment based translation model is the treatment of the bag of word translations. The word alignment based translation model generates a bag of words for each source word, while the chunk-based model constructs a set of target words from a set of source words. The behavior is modeled as a chunking procedure by first associating words to the head word of its chunk and then performing chunk-wise translation/insertion/deletion.

The complicated word alignment is handled by the determination of word positions in two stages: translation of chunk and chunk reordering. The former structures local orderings while the latter constitutes global orderings. In addition, the concept of head associated with each chunk plays the central role in constraining different levels of the reordering by the relative positions from heads.

3.3 Parameter Estimation

The parameter estimation for the chunk-based translation model relies on the EM-algorithm (Dempster et al., 1977). Given a large bilingual corpus the conditional probability of $P(\mathcal{J}, A, \mathcal{A}, \mathcal{E}|J, E) = P(J, \mathcal{J}, A, \mathcal{A}, \mathcal{E}|E)/\sum_{\mathcal{J},A,\mathcal{A},\mathcal{E}} P(J, \mathcal{J}, A, \mathcal{A}, \mathcal{E}|E)$ is first estimated for each pair of J and E (E-step), then each model parameters is computed based on the estimated conditional probability (M-step). The above procedure is iterated until the set of parameters converge.

However, this naive algorithm will suffer from severe computational problems. The enumeration of all possible chunkings $\mathcal{J}$ and $\mathcal{E}$ together with word alignment $\mathcal{A}$ and chunk alignment A requires a significant amount of computation. Therefore, we have introduced a variation of the Inside-Outside algorithm as seen in (Yamada and Knight, 2001) for E-step computation. The details of the procedure are described in Appendix A.

In addition to the computational problem, there exists a local-maximum problem, where the EM-Algorithm converges to a maximum solution but does not guarantee finding the global maximum. In order to solve this problem and to make the parameters converge quickly, IBM Model 4 parameters were used as the initial parameters for training. We directly applied the Lexicon Model and Fertility Model to the chunk-based translation model but set other parameters as uniform.

3.4 Decoding

The decoding algorithm employed for this chunk-based statistical translation is based on the beam search algorithm for word alignment statistical translation presented in (Tillmann and Ney, 2000), which generates outputs in left-to-right order by consuming input in an arbitrary order.

The decoder consists of two stages:

1. Generate possible output chunks for all possible input chunks.
2. Generate hypothesized output by consuming input chunks in arbitrary order and combining possible output chunks in left-to-right order.

The generation of possible output chunks is estimated through an inverted lexicon model and sequences of inserted strings (Tillmann and Ney, 2000). In addition, an example-based method is also introduced, which generates candidate chunks by looking up the viterbi chunking and alignment from a training corpus.

Since the combination of all possible chunks is computationally very expensive, we have introduced the following pruning and scoring strategies.

beam pruning: Since the search space is enormous, we have set up a size threshold to maintain partial hypotheses for both of the above two stages. We also incorporated a threshold for scoring, which allows partial hypotheses with a certain score to be processed.

example-based scoring: Input/output chunk pairs that appeared in a training corpus are "rewarded" so that they are more likely kept in the beam. During the decoding process, when a pair of chunks appeared in the first stage, the score is boosted by using this formula in the log domain,

$$\log P_{tm}(J|E) \quad + \quad \log P_{lm}(E) \quad + \quad weight \times \sum_j freq(\mathcal{E}_{A_j}, \mathcal{J}_j)$$

in which $P_{tm}(J|E)$ and $P_{lm}(E)$ are translation model and language model probability, respectively[1], $freq(\mathcal{E}_{A_j}, \mathcal{J}_j)$ is the frequency for the pair $\mathcal{E}_{A_j}$ and $\mathcal{J}_j$ appearing in the training corpus, and *weight* is a tuning parameter.

Table 1: Basic Travel Expression Corpus

	Japanese	English
# of sentences	171,894	
# of words	1,181,188	1,009,065
vocabulary size	20472	16232
# of singletons	82,06	5,854
3-gram perplexity	23.7	35.8

4 Experiments

The corpus for this experiment was extracted from the Basic Travel Expression Corpus (BTEC), a collection of conversational travel phrases for Japanese and English (Takezawa et al., 2002) as seen in Table 1. The entire corpus was split into three parts: 152,169 sentences for training, 4,846 sentences for testing, and the remaining 10,148 sentences for parameter tuning, such as the termination criteria for the training iteration and the parameter tuning for decoders.

Three translation systems were tested for comparison:

model4: Word alignment based translation model, IBM Model 4 with a beam search decoder.

chunk3: Chunk-based translation model, limiting the maximum allowed chunk size to 3.

model3+: chunk3 with example-based chunk candidate generation.

Figure 5 shows some examples of viterbi chunking and chunk alignment for chunk3.

Translations were carried out on 510 sentences selected randomly from the test set and evaluated according to the following criteria with 16 reference sets.

WER: Word-error-rate, which penalizes the edit distance against reference translations.

[1] For simplicity of notation, dependence on other variables are omitted, such as $\mathcal{J}$.

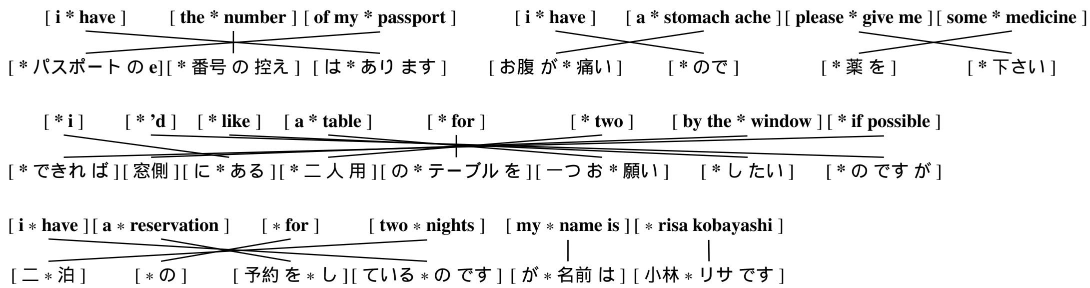

Figure 5: Examples of viterbi chunking and chunk alignment for English-to-Japanese translation model. Chunks are bracketed and the words with ∗ to the left are head words.

Table 2: Experimental results for Japanese–English translation

Model	WER [%]	PER [%]	BLEU [%]	SE [%] A	SE [%] A+B	SE [%] A+B+C
model4	43.3	37.2	46.5	59.2	74.1	80.2
chunk3	40.9	36.1	48.4	59.8	73.5	78.8
chunk3+	38.5	33.7	52.1	65.1	76.3	80.6

PER: Position independent WER, which penalizes without considering positional disfluencies.

BLEU: BLEU score, which computes the ratio of n-gram for the translation results found in reference translations (Papineni et al., 2002).

SE: Subjective evaluation ranks ranging from A to D (A:Perfect, B:Fair, C:Acceptable and D:Nonsense), judged by native speakers.

Table 2 summarizes the evaluation of Japanese-to-English translations, and Figure 6 presents some of the results by model4 and chunk3+.

As Table 2 indicates, chunk3 performs better than model4 in terms of the non-subjective evaluations, although it scores almost equally in subjective evaluations. With the help of example-based decoding, chunk3+ was evaluated as the best among the three systems.

input:	一 五 二 便 の 荷物 は これ で 全部 です か
reference:	is this all the baggage from flight one five two
model4:	is this all you baggage for flight one five two
chunk3:	is this all the baggage from flight one five two
input:	朝食 を ルームサービス で お 願い し ます
reference:	may i have room service for breakfast please
model4:	please give me some room service please
chunk3:	i 'd like room service for breakfast
input:	もしもし 三 月 十 九 日 の 予約 を 変更 し たい の です が
reference:	hello i 'd like to change my reservation for march nineteenth
model4:	i 'd like to change my reservation for ninety days be march hello
chunk3:	hello i 'd like to change my reservation on march nineteenth
input:	二 三 分 待っ て下さい 今 電話 中 な ん です
reference:	wait a couple of minutes i 'm telephoning now
model4:	is this the line is busy now a few minutes
chunk3:	i 'm on another phone now please wait a couple of minutes

Figure 6: Translation examples by word alignment based model and chunk-based model

5 Discussion

The chunk-based translation model was originally inspired by transfer-based machine translation but modeled by chunks in order to capture syntax-based correspondence. However, the structures evolved into complicated modeling: The translation model involves many stages, notably chunking and two kinds of reordering, word-based and chunk-based alignments. This is directly reflected in parameter estimation, where chunk3 took 20 days for 40 iterations, which is roughly the same amount of time required for training IBM Model 5 with pegging.

The unit of chunk in the statistical machine translation framework has been extensively discussed in the literature. Och et al. (1999) proposed a translation template approach that computes phrasal mappings from the viterbi alignments of a training corpus. Watanabe et al. (2002) used syntax-based phrase alignment to obtain chunks. Marcu and Wong (2002) argued for a different phrase-based translation modeling that directly induces a phrase-by-phrase lexicon model from word-wise data. All of these methods bias the training and/or decoding with phrase-level examples obtained by preprocessing a corpus (Och et al., 1999; Watanabe et al., 2002) or by allowing a lexicon model to hold phrases (Marcu and Wong, 2002). On the other hand, the chunk-based translation model holds the knowledge of how to construct a sequence of chunks from a sequence of words. The former approach is suitable for inputs with less deviation from a training corpus, while the latter approach will be able to perform well on unseen word sequences, although chunk-based examples are also useful for decoding to overcome the limited context of a n-gram based language model.

Wang (1998) presented a different chunk-based method by treating the translation model as a phrase-to-string process. Yamada and Knight (2001) further extended the model to a syntax-to-string translation modeling. Both assume that the source part of a translation model is structured either with a sequence of chunks or with a parse tree, while our method directly models a string-to-string procedure. It is clear that the string-to-string modeling with hiden chunk-layers is computationally more expensive than those structure-to-string models. However, the structure-to-string approaches are already biased by a monolingual chunking or parsing, which, in turn, might not be able to uncover the bilingual phrasal or syntactical constraints often observed in a corpus.

Alshawi et al. (2000) also presented a two-level arranged word ordering and chunk ordering by a hierarchically organized collection of finite state transducers. The main difference from our work is that their approach is basically deterministic, while the chunk-based translation model is non-deterministic. The former method, of course, performs more efficient decoding but requires stronger heuristics to generate a set of transducers. Although the latter approach demands a large amount of decoding time and hypothesis space, it can operate on a very broad-coverage corpus with appropriate translation modeling.

Acknowledgments

The research reported here was supported in part by a contract with the Telecommunications Advancement Organization of Japan entitled "A study of speech dialogue translation technology based on a large corpus".

References

Hiyan Alshawi, Srinivas Bangalore, and Shona Douglas. 2000. Learning dependency translation models as collections of finite state head transducers. *Computational Linguistics*, 26(1):45–60.

Peter F. Brown, 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.

A. P. Dempster, N.M. Laird, and D.B.Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, B(39):1–38.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proc. of EMNLP-2002*, Philadelphia, PA, July.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. of EMNLP/WVLC*, University of Maryland, College Park, MD, June.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL 2002*, pages 311–318.

Toshiyuki Takezawa, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto. 2002. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proc. of LREC 2002*, pages 147–152, Las Palmas, Canary Islands, Spain, May.

Christoph Tillmann and Hermann Ney. 2000. Word re-ordering and dp-based search in statistical machine translation. In *Proc. of the COLING 2000*, July-August.

Ye-Yi Wang. 1998. *Grammar Inference and Statistical Machine Translation*. Ph.D. thesis, School of Computer Science, Language Technologies Institute, Carnegie Mellon University.

Taro Watanabe, Kenji Imamura, and Eiichiro Sumita. 2002. Statistical machine translation based on hierarchical phrase alignment. In *Proc. of TMI 2002*, pages 188–198, Keihanna, Japan, March.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proc. of ACL 2001*, Toulouse, France.

Appendix A Inside-Outside Algorithm for Chunk-based Translation Model

The basic idea of inside-outside computation is to separate the whole process into two parts, chunk translation and chunk reordering. Chunk translation handles translation of each chunk, while chunk reordering performs chunking and chunk reoprdering. The inside (backward or beta) probabilities can be derived, which represent the probability of source/target paring of chunks and sentences. The outside (forward or alpha) probabilities can be defined as the probability of a particular source and target pair appearing at a particular chunking and reordering.

Inside Probability First, given E and J, compute chunk translation inside probabilities for all the possible source and target chunks pairing $E_i^{i'}$ and $J_j^{j'}$ in which $E_i^{i'}$ is the chunk ranging from index i to i',

$$\begin{aligned}\beta(E_i^{i'}, J_j^{j'}) &= \sum_{A'} P(A, J_j^{j'} | E_i^{i'}) \\ &= \sum_{A'} \prod_{\theta} P_\theta(A', J_j^{j'}, E_i^{i'})\end{aligned}$$

where P_θ is the probability of a model with associated values for corresponding random variables, such as $\epsilon(\varphi_i|E_i)$ or $\tau(J_j|E_i)$, except for the chunk reorder model ϱ. A' is a word alignment for the chunks $E_i^{i'}$ and $J_j^{j'}$.

Second, compute the inside probability for sentence pairs E and J by considering all possible chunkings and chunk alignments.

$$\begin{aligned}\beta(E, J) &= \sum_{\mathcal{E},\mathcal{J}:|\mathcal{E}|=|\mathcal{J}|} \sum_{A} P(A, \mathcal{E}, \mathcal{J}, J|E) \\ &= \sum_{\mathcal{E},\mathcal{J}:|\mathcal{E}|=|\mathcal{J}|} \sum_{A} P(A|\mathcal{E}, \mathcal{J}) \prod_{j} \beta(\mathcal{E}_{A_j}, \mathcal{J}_j)\end{aligned}$$

Outside Probability The outside probability for sentence pairing is always 1.

$$\alpha(E, J) = 1.0$$

The outside probabilities for each chunk pair is

$$\begin{aligned}\alpha(E_i^{i'}, J_j^{j'}) &= \alpha(E, J) \sum_{\mathcal{E},\mathcal{J}:|\mathcal{E}|=|\mathcal{J}|} \sum_{A} P(A|\mathcal{E}, \mathcal{J}) \\ &\times \prod_{\mathcal{E}_{A_k} \neq E_i^{i'}, \mathcal{J}_k \neq J_j^{j'}} \beta(\mathcal{E}_{A_k}, \mathcal{J}_k)\end{aligned}$$

.

Inside-Outside Computation The combination of the above inside-outside probabilities yields the following formulas for the accumulated counts of pair occurrences.

First, the counts for each model parameter θ with associated random variables $count_\theta(\Theta)$ is

$$\begin{aligned}count_\theta(\Theta) &= \sum_{<E,J>} \sum_{\Theta(A', E_i^{i'}, J_j^{j'})} \alpha(E_i^{i'}, J_j^{j'}) / \beta(E, J) \\ &\times \prod_{\theta'} P_{\theta'}(A', J_j^{j'}, E_i^{i'})\end{aligned}$$

.

Second, the count for chunk reordering with associated random variables $count_\varrho(\Theta)$ is

$$\begin{aligned}count_\varrho(\Theta) &= \sum_{<E,J>} \alpha(E, J) / \beta(E, J) \\ & \sum_{\Theta(A,\mathcal{E},\mathcal{J})} P_\varrho(A|\mathcal{E}, \mathcal{J}) \prod_{k} \beta(\mathcal{E}_{A_k}, \mathcal{J}_k)\end{aligned}$$

.

Approximation Even with the introduction of the inside-outside parameter estimation paradigm, the enumeration of all possible chunk pairing and word alignment requires $O(lmk^4(k+1)^k)$ computations, where l and m are sentence length for E and J, respectively, and k is the maximum allowed number of words per chunk. In addition, the enumeration of all possible alignments for all possible chunked sentences is $O(2^l 2^m n!)$, where $n = |\mathcal{J}| = |\mathcal{E}|$.

In order to handle the massive amount of computational demand, we have applied an approximation to the inside-outside estimation procedure. First, the enumeration of word alignment computation for chunk translations was approximated by a set of alignments, the viterbi alignment and neighboring alignment through move/swap operations of particular word alignments.

Second, the chunk alignment enumeration was also approximated by a set of chunking and chunk alignments as follows.

1. Determines the number of chunks per sentence
2. Determine initial chunking and alignment
3. Compute viterbi chunking-alignment via hill-climbing using the following operators
   - Move boundary of chunk
   - Swap chunk alignment
   - Move head position
4. Compute neighboring chunking-alignment using the above operators