Bidirectional Decoding for Statistical Machine Translation

Taro WATANABE †‡ and **Eiichiro SUMITA** †
{taro.watanabe, eiichiro.sumita}@atr.co.jp
† ATR Spoken Language Translation Research Laboratories
2-2-2 Hikaridai Seika-cho,
Soraku-gun, Kyoto 619-0288 JAPAN
‡ Department of Information Science
Kyoto University
Sakyo-ku, Kyoto 606-8501, JAPAN

Abstract

This paper describes the right-to-left decoding method, which translates an input string by generating in right-to-left direction. In addition, presented is the bidirectional decoding method, that can take both of the advantages of left-to-right and right-to-left decoding method by generating output in both ways and by merging hypothesized partial outputs of two directions. The experimental results on Japanese and English translation showed that the right-to-left was better for Englith-to-Japanese translation, while the left-to-right was suitable for Japanese-to-English translation. It was also observed that the bidirectional method was better for English-to-Japanese translation.

1 Introduction

The statistical approach to machine translation regards the machine translation problem as the maximum likelihood solution of a translation target text given a translation source text. According to the Bayes Rule, the problem is transformed into the noisy channel model paradigm, where the translation is the maximum a posteriori solution of a distribution for a channel target text given a channel source text and a prior distribution for the channel source text (Brown et al., 1993).

Although there exists efficient algorithms to estimate the parameters for the statistical machine translation (SMT), one of the problems of SMT is the search algorithms for the translation given a sequence of words. There exists stack decoding algorithm (Berger et al., 1996), A* search algorithm (Och et al., 2001; Wang and Waibel, 1997) and dynamic-programming algorithms (Tillmann and Ney, 2000; Garcia-Varea and Casacuberta, 2001), and all translate a given input string word-by-word and render the translation in left-to-right, with pruning technologies assuming almost linearly aligned translation source and target texts. The algorithms proposed above cannot deal with drastically different word correspondence, such as Japanese and English translation, where Japanese is SOV while SVO in English. Germann et al. (2001) suggested greedy method and integer programming decoding, though the first method suffer from the similar problem as described above and the second is impractical for the real-world application.

This paper presents two decoding methods, one is the right-to-left decoding based on the left-to-right beam search algorithm, which generates outputs from the end of a sentence. The second one is the bidirectional decoding method which decodes in both of the left-to-right and right-to-left directions and merges the two hypothesized partial sentences into one. The experimental results of Japanese and English translation indicated that the right-to-left decoding was better for English-to-Japanese translation, while the left-to-right decoding was better for Japanese-to-English decoding. The above results could be justified by the structural difference of Japanese and English, where English takes the prefix structure that places emphasis at the beginning of a sentence, hence prefers left-to-right decoding. On the other hand, Japanese takes postfix structure, setting attention around the end of a sentence, therefore favors right-to-left decoding. The bidirectional decoding, which can take both of the benefits of decoding method, was superior to mono-directional decoding methods.

The next section briefly describes the SMT focusing on the IBM Model 4. Then, the Section 3 presents decoding algorithms in three direction, left-to-right, right-to-left and bi-direction. The Section 4 presents the results of Japanese and English translation followed by discussions.

2 Statistical Machine Translation

Statistical machine translation regards machine translation as a process of translating a source lan-

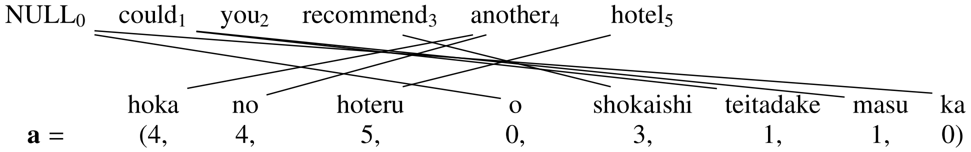

NULL0 could1 you2 recommend3 another4 hotel5
hoka no hoteru o shokaishi teitadake masu ka
a = (4, 4, 5, 0, 3, 1, 1, 0)

Figure 1: An example of alignment for Japanese and English sentences

guage text (**f**) into a target language text (**e**) with the following formula:

$$\mathbf{e} = \arg\max_{\mathbf{e}} P(\mathbf{e}|\mathbf{f})$$

The Bayes Rule is applied to the above to derive:

$$\mathbf{e} = \arg\max_{\mathbf{e}} P(\mathbf{f}|\mathbf{e})P(\mathbf{e})$$

The translation process is treated as a noisy channel model, like those used in speech recognition in which there exists **e** transcribed as **f**, and a translation is to infer the best **e** from **f** in terms of $P(\mathbf{f}|\mathbf{e})P(\mathbf{e})$. The former term, $P(\mathbf{f}|\mathbf{e})$, is a translation model representing some correspondence between bilingual text. The latter, $P(\mathbf{e})$, is the language model denoting the likelihood of the channel source text. In addition, a word correspondence model, called alignment **a**, is introduced to the translation model to represent a positional correspondence of the channel target and source words:

$$\mathbf{e} = \arg\max_{\mathbf{e}} \sum_{\mathbf{a}} P(\mathbf{f}, \mathbf{a}|\mathbf{e})P(\mathbf{e})$$

An example of an alignment is shown in Figure 1, where the English sentence “could you recommend another hotel” is mapped onto the Japanese “hoka no hoteru o shokaishi teitadake masu ka”, and both “hoka” and “no” are aligned to “another”, etc. The NULL symbol at index 0 is also a lexical entry in which no morpheme is aligned from the channel target morpheme, such as “masu” and “ka” in this Japanese example.

2.1 IBM Model 4

The IBM Model 4, main focus in this paper, is composed of the following models (see Figure 2):

- Lexical Model — $t(f|e)$: Word-for-word translation model, representing the probability of a source word f being translated into a target word e.
- Fertility Model — $n(\phi|e)$: Representing the probability of a source word e generating ϕ words.
- Distortion Model — d : The probability of distortion. In Model 4, the model is decomposed into two sets of parameters:
  - $d_1(j - c_{\rho i}|\mathcal{A}(e_i), \mathcal{B}(f_j))$: Distortion probability for head words. The head word is the first of the target words generated from a source word a cept, that is the channel source word with fertility more than and equal to one. The head word position j is determined by the word classes of the previous source word, $\mathcal{A}(e_i)$, and target word, $\mathcal{B}(f_j)$, relative to the centroid of the previous source word, c_{ρ_i}.
  - $d_{>1}(j - j'|\mathcal{B}(f_j))$: Distortion probability for non-head words. The position of a non-head word j is determined by the word class and relative to the previous target word generated from the cept (j').
- NULL Translation Model — p_1 : A fixed probability of inserting a NULL word after determining each target word f.

For details, refer to Brown et al. (1993).

2.2 Search Problem

The search problem of statistical machine translation is to induce the maximum likely channel source sequence, **e**, given **f** and the model, $P(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} P(\mathbf{f}, \mathbf{a}|\mathbf{e})$ and $P(\mathbf{e})$. For the space of **a** is extremely large, $|\mathbf{a}|^{l+1}$, where the l is the output length, an approximation of $P(\mathbf{f}|\mathbf{e}) \simeq P(\mathbf{f}, \mathbf{a}|\mathbf{e})$ is used when exploring the possible candidates of translation.

This problem is known to be NP-Complete (Knight, 1999), for the re-ordering property in the model further complicates the search. One of the solution is the left-to-right generation of output by consuming input words in any-order. Under this constraint, many researchers had contributed algorithms and associated pruning strategies, such as Berger et al. (1996), Och et al. (2001), Wang and Waibel (1997), Tillmann and Ney (2000) Garcia-Varea and Casacuberta (2001) and Germann et al. (2001), though they all based on almost linearly

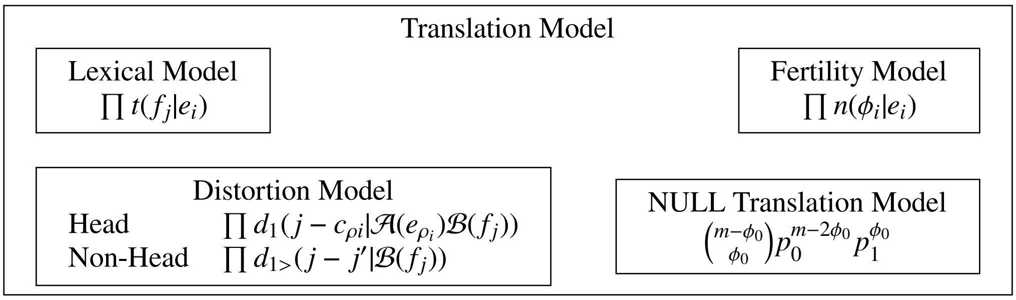

Translation Model
Lexical Model
$\prod t(f_j|e_i)$
Fertility Model
$\prod n(\phi_i|e_i)$
Distortion Model
Head $\prod d_1(j - c_{\rho i}|\mathcal{A}(e_{\rho_i})\mathcal{B}(f_j))$
Non-Head $\prod d_{1>}(j - j'|\mathcal{B}(f_j))$
NULL Translation Model
$\binom{m-\phi_0}{\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0}$

Figure 2: Translation Model (IBM Model 4)

aligned language pairs, and not suitable for language pairs with totally different alignment correspondence, such as Japanese and English.

3 Decoding Algorithms

The decoding methods presented in this paper explore the partial candidate translation hypotheses greedily, as presented in Tillmann and Ney (2000) and Och et al. (2001), and operation applied to each hypothesis is similar to those explained in Berger et al. (1996), Och et al. (2001) and Germann et al. (2001). The algorithm is depicted in Algorithm 1 where $C = \{j_k : k = 1...|C|\}$ represents a set of input string position [1]. The algorithm assumes two kinds of partial hypotheses[2], translated partially from an input string, one is an open hypothesis that can be extended by raising the fertility. The other is a close hypothesis that is to be extended by inserting a string $\mathbf{e}'$ to the hypothesis. The $\mathbf{e}'$ is a sequence of output word, consisting of a word with the fertility more than one (translation of f_j) and other words with zero fertility. The translation of f_j can be computed either by inverse translation table (Och et al., 2001; Al-Onaizan et al., 1999). The list of zero fertility words can be obtained from the viterbi alignment of training corpus (Germann et al., 2001). The extension operator applied to an open hypothesis $(\mathbf{e}, C)$ is:

- *align j to e_i* — this creates a new hypothesis by raising the fertility of e_i by consuming the input word f_j. The generated hypothesis can be treated as either closed or open, that means to stop raising the fertility or raise the fertility further more.

The operators applied to a close hypothesis are:

Algorithm 1 Beam Decoding Search

```
input  source string: $f_1 f_2 ... f_m$
  for all cardinality $c = 0, 1, ...m - 1$ do
    for all $(\mathbf{e}, C)$ where $|C| = c$ do
      for all $j = 1, ...m$ and $j \notin C$ do
        if $(\mathbf{e}, C)$ is open then
          align $j$ to $e_i$ and keep it open
          align $j$ to $e_i$ and close it
        else
          align $j$ to NULL
          insert $\mathbf{e}'$, align from $j$ and open it
          insert $\mathbf{e}'$, align from $j$ and close it
        end if
      end for
    end for
  end for
```

- *align j to NULL* — raise the fertility for the NULL word.

- *insert* $\mathbf{e}'$*, align from j* — this operator insert a string $\mathbf{e}'$ and align one input word f_j to one of the word in $\mathbf{e}'$. After this operation, the new hypothesis can be regarded as either open or closed.

Pruning is inevitable in the process of decoding, and applied is the beam search pruning, in which the maximum number of hypotheses to be considered is limited. In addition, fertility pruning is also introduced which suppress the word with large number of fertility. The skipping based criteria, such as introduced by Och et al. (2001), is not appropriate for the language pairs with drastically different alignment, such as Japanese and English, hence was not considered in this paper. Depending on the output generation direction, the algorithm can generate either in left-to-right or right-to-left, by alternating some constraints of insertion of output words.

[1] For simplicity, the dependence of alignment, **a** is omitted.

[2] There exist a complete hypothesis, that is a candidate of translation.

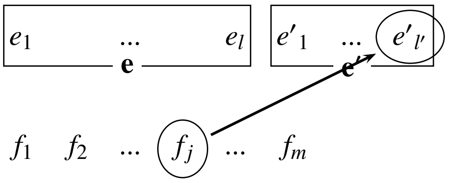


Figure 3: string insertion operator for left-to-right decoding method. A string $\mathbf{e}'$ was appended after the partial output string, $\mathbf{e}$, and the last word in $\mathbf{e}'$ was aligned from f_j.

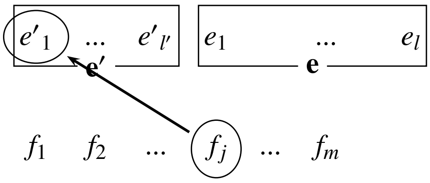


Figure 4: string insertion operation for right-to-left decoding method. A string $\mathbf{e}'$ was prepended before the partial output string, $\mathbf{e}$, and the first word in $\mathbf{e}'$ was aligned from f_j.

3.1 Left-to-Right Decoding

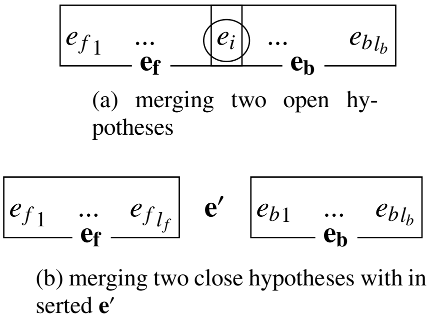
The left-to-right decoding enforces the restriction where the insertion of $\mathbf{e}'$ is allowed after the partially generated $\mathbf{e}$, and alignment from the input word f_j is restricted to the end of the word of $\mathbf{e}'$. Hence, the operator applied to an open hypothesis raise the fertility for the word at the end of $\mathbf{e}$ (refer to Figure 3).

The language which place emphasis around the beginning of a sentence, such as English, will be suitable in this direction, for the Language Model score $P(\mathbf{e})$ can estimate what should come first. Hence, the decoder can discriminate a hypothesis better or not.

3.2 Right-to-Left Decoding

The right-to-left decoding does the reverse of the left-to-right decoding, in which the insertion of $\mathbf{e}'$ is allowed only before the $\mathbf{e}$ and the f_j is aligned to the beginning of the word of $\mathbf{e}'$ (see Figure 4). Therefore, the open hypothesis is extended by raising the fertility of the beginning of the word of $\mathbf{e}$. In prepending a string to a partial hypothesis, an alignment vector should be reassigned so that the values can point out correct index.

Again, the right-to-left direction is suitable for the language which enforces stronger constraints at the end of sentence, such as Japanese, similar to the reason mentioned above.



(a) merging two open hypotheses

(b) merging two close hypotheses with inserted $\mathbf{e}'$

Figure 5: Merging left-to-right and right-to-left hypotheses ($\mathbf{e_f}$ and $\mathbf{e_b}$) in bidirectional decoding method. Figure 5(a) merge two open hypotheses, while Figure 5(b) merge them with inserted zero fertility words.

3.3 Bidirectional Decoding

The bidirectional decoding decode the input words in both direction, one with left-to-right decoding method up to the cardinality of $\lceil m/2 \rceil$ and right-to-left direction up to the cardinality of $\lfloor m/2 \rfloor$, where m is the input length. Then, the two hypotheses are merged when both are open and can share the same output word e, which resulted in raising the fertility of e. If both of them are closed hypotheses, then an additional sequence of zero fertility words (or NULL sequence) are inserted (refer to Figure 5).

3.4 Computational Complexity

The computational complexity for the left-to-right and right-to-left is the same, $O(|E|^3 m^2 2^m)$, as reported by Tillmann and Ney (2000), in which $|E|$ is the size of the vocabulary for output sentences [3]. The bidirectional method involves merging of two hypotheses, hence additional $O(\binom{m}{m/2})$ is required.

3.5 Effects of Decoding Direction

The decoding algorithm generating in left-to-right direction fills the output sequence from the beginning of a sentence by consuming the input words in any order and by selecting the corresponding translation.

Therefore, the languages with prefix structure, such as English, German or French, can take the benefits of this direction, because the language model/translation model can differentiate "good" hypotheses to "bad" hypotheses around the beginning of the output sentences. Therefore, the narrowing the search space by the beam search crite-

[3] The term $|E|^3$ is the case for trigram language model.

ria (pruning) would not affect the overall quality. On the other hand, if right-to-left decoding method were applied to such a language above, the difference of good hypotheses and bad hypotheses is small, hence the drop of hypotheses would affect the quality of translation.

The similar statement can hold for postfix languages, such as Japanese, where emphasis is placed around the end of a sentence. For such languages, right-to-left decoding will be suitable but left-to-right decoding will degrade the quality of translation.

The bidirectional decoding is expected to take the benefits of both of the directions, and will show the best results in any kind of languages.

4 Experimental Results

The corpus for this experiment consists of 172,481 bilingual sentences of English and Japanese extracted from a large-scale travel conversation corpus (Takezawa et al., 2002). The statistics of the corpus are shown in Table 1. The database was split into three parts: a training set of 152,183 sentence pairs, a validation set of 10,148, and a test set of 10,150.

Table 1: Statistics on a travel conversation corpus

	Japanese	English
# of sentences	172,481	
# of words	1,186,620	1,005,080
vocabulary size	22,801	15,768
avg. sentence length	6.88	5.83
3-gram perplexity	26.16	36.92

The translation models, both for the Japanese-to-English (J-E) and English-to-Japanese (E-J) translation, were trained toward IBM Model 4 on the training set and cross-validated on validation set to terminate the iteration by observing perplexity. In modeling IBM Model 4, POSs were used as word classes.

From the viterbi alignments of the training corpus, A list of possible insertion of zero fertility words were extracted with frequency more than 10, around 1,300 sequences of words for both of the J-E and E-J translations. The test set consists of 150 Japanese sentences varying by the sentence length of 6, 8 and 10. The translation was carried out by three decoding methods:left-to-right, right-to-left and bidirectional one.

The translation results were evaluated by word-error-rate (WER) and position independent word-error-rate (PER) (Watanabe et al., 2002; Och et al., 2001). The WER is the measure by penalizing insertion/deletion/replacement by 1. The PER is the one similar to WER but ignores the positions, allowing the reordered outputs, hence can estimate the accuracy for the tranlslation word selection. It has been also evaluated by subjective evaluation (SE) with the criteria ranging from A(perfect) to D(nonsense) [4] (Sumita et al., 1999).

Table 3: Comparison of the three decoders by the ratio each decoder produced search errors.

	J-E	E-J
LtoR	11.3	12.0
RtoL	59.3	34.0
Bi	15.3	15.3

Table 2 summarizes the results of decoding by left-to-right, right-to-left and bidirectional method evaluated with WER, PER and SE. Table 3 shows the ratio of producing search errors, computed by comparing the translation model and lnguage model scores for the outputs from three decoding methods. Sample Japanese-to-English translations performed by the decoders is presented in Figure 6.

5 Discussions

From Table 2, the left-to-right decoding method performed better than the right-to-left one in Japanese-to-English translation as expected in Section 3.5. Furthermore, the bidirectional decoding method was slightly better than the left-to-right one, for it could combine the benefits of both directions.

Similar analysis could hold for English-to-Japanese translation, and the right-to-left decoding method was slightly superior to the left-to-right one in terms of WER/PER scores, though the SE score dropped from 8.7% to 6.7% in C-ranked sentences. Overall quality measured by the SE rate for accepted senteces, ranging from A to C, dropped from 68.0% into 66.0%. In addition, the bidirectional method in English-to-Japanese translation was not evaluated as high as those in Japanese-to-English translation: the results were closer to the left-to-right method. This might be due to the nature of lan-

[4] The meanings of the symbol are follows: A — perfect: no problem in either information or grammar; B — fair: easy to understand but some important information is missing or it is grammatically flawed; C — acceptable: broken but understandable with effort; D — nonsense: important information has been translated incorrectly.

Table 2: Summary of results for Japanese-to-English (J-E) and English-to-Japanese (E-J) translations by left-to-right (LtoR), right-to-left (RtoL) and bidirectional (Bi) decoding methods.

Trans.	Alg.	WER	PER	SE			
				A	B	C	D
J-E	LtoR	70.0	64.8	26.7%	23.3%	20.0%	30.0%
	RtoL	74.6	66.9	21.3%	24.7%	18.0%	36.0%
	Bi	69.9	63.7	27.3%	22.7%	20.7%	29.3%
E-J	LtoR	66.2	57.6	49.3%	10.0%	8.7%	32.0%
	RtoL	64.0	56.1	49.3%	10.0%	6.7%	34.0%
	Bi	66.0	58.0	48.7%	8.0%	10.0%	33.3%

input:	suri ni saifu o sura re mashi ta
	(i had my pocket picked)
LtoR:	here 's my wallet was stolen
RtoL:	here 's my wallet was stolen
Bi:	i had my wallet stolen
input:	sumimasen ga terasu no seki ga ii no desu ga
	(excuse me but can we have a table on the terrace)
LtoR:	excuse me i 'd like a seat on the terrace
RtoL:	i 'd prefer excuse me
Bi:	i 'd like a seat on the terrace
input:	nan ji ni owaru no desu
	(what time will it be over)
LtoR:	what time should i be at the end
RtoL:	it 's what time will it be over
Bi :	at what time is it end
input:	nimotsu o ue ni age te morae masu ka
	(will you put my luggage on the rack)
LtoR:	could you put my baggage here
RtoL:	do you have overhead luggage
Bi:	could you put my baggage
input:	ee ani to imouto ga hitori zutsu i masu
	(yes i have a brother and a sister)
LtoR:	yes brother and sister there a daughter
RtoL:	you 're yes brother and sister daughter
Bi:	yes my daughter is there a brother and sister

Figure 6: Examples of Japanese-to-English translation

guage model employed for this experiment, for the language model probabilities were assigned based on the left history, not the right history. It is expected that the use of the suitable language model context direction corresponding to a generation direction would assign appropriate probability, hence would be able to differentiate better hypotheses.

Table 3 indicats that the right-to-left decoding method produced more errors than other methods regardless of translaiton directions. This is explained by the use of the left history language model, not the right context one, as stated above. Nevertheless, the search error decreased from 59.3 into 34.0 by alternating the translation direction for the right-to-left decoding method, which still supports the use of the correct rendering direction for translation target language.

6 Conclusion

The decoding methods for statistical machine translation presented here varies the output directions, left-to-right, right-to-left and bi-direction, and were experimented with drastically different language pairs, English and Japanese. The results indicated that the left-to-right decoding method was suitable for Japanese-to-English translation while the right-to-left decoding method fit with English-to-Japanese translation. In addition, the bidirectional decoding method was superior to mono-directional decoding method for Japanese-to-English translation. This suggests that the translation output generation should match with the underlying linguistic structure for the output language.

Acknowledgement

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

Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Frantz-Josef Och, David Purdy, Noah A. Smith, and David Yarowsky. 1999. Statistical machine translation final report, jhu workshop 1999, 12.

A. Berger, P. Brown, S. Pietra, V. Pietra, J. Gillett, A. Kehler, and R. Mercer. 1996. Language translation apparatus and method of using context-based translation models. Technical report, United States Patent, Patent Number 5510981, April.

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.

Ismael Garcia-Varea and Francisco Casacuberta. 2001. Search algorithms for statistical machine translation based on dynamic programming and pruning techniques. In *MT Summit VIII*, Santiago de Compostela, Galicia, Spain, september.

Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. 2001. Fast decoding and optimal decoding for machine translation. In *Proc. of ACL-01*, Toulouse, France.

Kevin Knight. 1999. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

Franz Josef Och, Nicola Ueffing, and Hermann Ney. 2001. An efficient a* search algorithm for statistical machine translation. In *Proc. of the ACL-2001 Workshop on Data-Driven Machine Translation*, pages 55–62, Toulouse, France, July.

Eiichiro Sumita, Setsuo Yamada, Kazuhide Yamamoto, Michael Paul, Hideki Kashioka, Kai Ishikawa, and Satoshi Shirai. 1999. Solutions to problems inherent in spoken-language translation: The ATR-MATRIX approach. In *Machine Translation Summit VII*, pages 229–235.

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 and Alex Waibel. 1997. Decoding algorithm in statistical machine translation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*.

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