Integrated Chinese Word Segmentation in Statistical Machine Translation

Jia Xu and *Evgeny Matusov* and *Richard Zens* and *Hermann Ney*

Lehrstuhl für Informatik VI - Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
{xujia,matusov,zens,ney}@cs.rwth-aachen.de

Abstract

A Chinese sentence is represented as a sequence of characters, and words are not separated from each other. In statistical machine translation, the conventional approach is to segment the Chinese character sequence into words during the pre-processing. The training and translation are performed afterwards. However, this method is not optimal for two reasons: 1. The segmentations may be erroneous. 2. For a given character sequence, the best segmentation depends on its context and translation.

In order to minimize the translation errors, we take different segmentation alternatives instead of a single segmentation into account and integrate the segmentation process with the search for the best translation. The segmentation decision is only taken during the generation of the translation. With this method we are able to translate Chinese text at the character level. The experiments on the IWSLT 2005 task showed improvements in the translation performance using two translation systems: a phrase-based system and a finite state transducer based system. For the phrase-based system, the improvement of the BLEU score is 1.5% absolute.

1. Introduction

In Chinese texts, words composed of single or multiple characters are not separated by white space, which is different from most of the European languages.

In statistical machine translation, the conventional way is to segment the Chinese character sequence into Chinese words before the training and translation.

We compared different segmentation methods in [1]. The training and test texts can be segmented into words or used at the character level. In the experiments in [1], the translation results with the previous method outperformed the results with the latter one.

Here we continued the investigation on the translation of the text at the character level and developed a new method yielding better translation results than when translation is at the word level.

This method handles all the segmentation alternatives instead of only the single-best segmentation. The single-best one may contain errors or may be not optimal with respect to the training corpus.

Instead of reading a single best segmented sentence, our system handles all the segmentation alternatives by reading a segmentation lattice. Similar approaches were applied in the speech translation, e.g. [2], where the speech recognition and text translation are combined by using the recognition lattices. We also weight the different segmentations with a language model trained on the Chinese corpus at the word level. Weighting the word segmentation by language model cost was introduced in [3].

To verify the improvements with the integrated segmentation method, we experimented on two translation systems: translation with the weighted finite state transducers and translation with the phrase based approach. On the IWSLT 2005 task [4], using a phrase-based translation system, the improvement of the BLEU score reached 1.5% absolute.

This paper is structured as follows: first we will briefly review the baseline statistical machine translation system in Section 2. In Section 3 we will discuss the idea, the theory, as well as the generation process of the integrated segmentation approach compared to the conventional approach. The experimental results for the IWSLT 2005 task [4] will be presented in Section 4.

2. Statistical machine translation system

2.1. Bayes decision rule

In statistical machine translation, we are given a source language sentence $f_1^J = f_1 \dots f_j \dots f_J$, which is to be translated into a target language sentence $e_1^I = e_1 \dots e_i \dots e_I$. Among all possible target language sentences, we will choose the sentence with the highest probability:

$$\hat{e}_1^{\hat{I}} = \underset{e_1^I, I}{\operatorname{argmax}} \left\{ Pr(e_1^I | f_1^J) \right\} \quad (1)$$

$$= \underset{e_1^I, I}{\operatorname{argmax}} \left\{ Pr(e_1^I) \cdot Pr(f_1^J | e_1^I) \right\} \quad (2)$$

The decomposition into two knowledge sources in Equation 2 is known as the source-channel approach to statistical machine translation [5]. It allows an independent model-

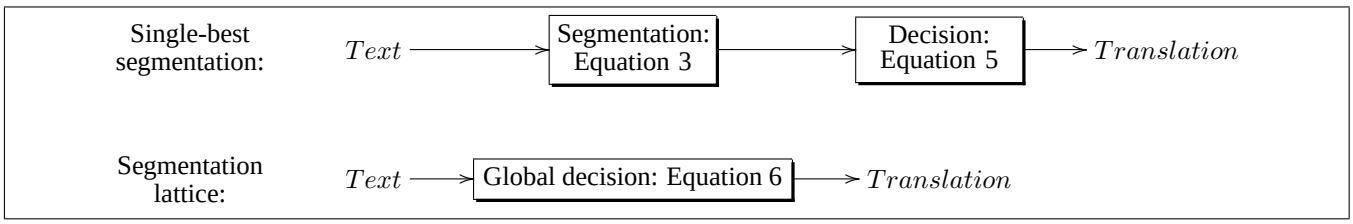

Figure 1: Segmentation methods

ing of the target language model $Pr(e_1^I)$ and the translation model $Pr(f_1^J|e_1^I)$[1].

In our system, the translation model is trained on a bilingual corpus using GIZA++ [6], and the language model is trained with the SRILM toolkit [7].

2.2. Weighted finite-state transducer-based translation

We use the weighted finite-state tool by [8]. A *weighted finite-state transducer* $(Q, \Sigma \cup \{\epsilon\}, \Omega \cup \{\epsilon\}, K, E, i, F, \lambda, \rho)$ is a structure with a set of states Q, an alphabet of input symbols Σ, an alphabet of output symbols Ω, a weight semiring K, a set of arcs E, a single initial state i with weight λ and a set of final states F weighted by the function $\rho : F \rightarrow K$. A *weighted finite-state acceptor* is a weighted finite-state transducer without the output alphabet.

A composition algorithm is defined as: Let $T_1 : \Sigma^* \times \Omega^* \rightarrow K$ and $T_2 : \Omega^* \times \Gamma^* \rightarrow K$ be two transducers defined over the same semiring K. Their composition $T_1 \circ T_2$ realizes the function $T : \Sigma^* \times \Gamma^* \rightarrow K$.

By using the structure of the weighted finite-state transducers, the translation model is simply estimated as the language model on a bilanguage of source phrase/target phrase tuples, see [9].

2.3. Phrase-based translation

The phrase-based translation model is described in [10]. A phrase is a contiguous sequence of words. The pairs of source and target phrases are extracted from the training corpus and used in the translation.

The phrase translation probability $Pr(e_1^I|f_1^J)$ is modeled directly using a weighted log-linear combination of a trigram language model and various translation models: a phrase translation model and a word-based lexicon model. These translation models are used for both directions: $p(f|e)$ and $p(e|f)$. Additionally, we use a word penalty and a phrase penalty. The model scaling factors are optimized with respect to some evaluation criterion [11].

[1]The notational convention will be as follows: we use the symbol $Pr(\cdot)$ to denote general probability distributions with (nearly) no specific assumptions. In contrast, for model-based probability distributions, we use the generic symbol $p(\cdot)$.

3. Segmentation methods

3.1. Conventional segmentation methods

In this section, we give a short overview of the current Chinese word segmentation methods in statistical machine translation, most of these methods can be classified into three categories:

- The training and test texts are segmented with an automatic segmentation tool.

  Many segmentation tools use the dynamic programming algorithm and find the word boundaries which maximize the product of the word frequencies. But the segmentation may contain some errors, and we also found that a much more accurate word segmentation does not always lead to a large improvement in the translation performance.

- The training and test texts are segmented manually.

  Manual segmentation avoids segmentation errors but requires a human effort. Moreover, the correct segmentation will not result in the best translation result, if the segmentations in the test and training sets are inconsistent.

- Each Chinese character is treated as a word

  Training and translation at the Chinese character level do not require additional tool or human effort. But [1] showed that the translation results are not so good as the results obtained when translation is at the word level.

To minimize the number of lexicon entries and to ensure the consistency of the segmentations in the training and in the translation, we developed a new segmentation method, which uses the training text at the word level and translate the test text at the character level.

3.2. Idea

Figure 1 shows the translation procedures. With the conventional method, only a single-best word segmentation is transferred to the search for the best translation. This approach is not ideal because the segmentation may not be optimal for the translation. Taking hard decisions in word segmentation may lead to loss of the correct Chinese words.

Table 1: Example of a sentence and its translations.

Source sentence in characters:	zai na li ban li deng ji shou xu ?
Manually segmented source sentence:	zai nali banli dengji shouxu ?
Translation by single-best segmentation:	where to go through boarding formalities ?
Translation by segmentation lattice:	where do i make my boarding arrangements ?
One reference:	where do i complete boarding procedures ?

With the integrated segmentation method in Figure 1, for one input sentence, we take different segmentation alternatives into account and represent them as a lattice. The input to the translation system is then a set of lattices instead of the segmented text. The search decision of the word segmentation is therefore combined with the translation decision, and the best segmentation of a sentence is only selected while the translation is generated.

3.3. Theory

In this section, we will explain the methods in Figure 1 in detail. First, we will describe a general word segmentation model and then how it is used as a single-best segmentation or as a segmentation lattice.

A Chinese input sentence is denoted here as c_1^K at the character level and f_1^J at the word level, where $c_1 \dots c_k \dots c_K$ are the succeeding characters and $f_1 \dots f_j \dots f_J$ are the succeeding words.

Word segmentation model

The best segmented Chinese sentence $\hat{f}_1^{\hat{J}}$ with $\hat{J}$ words can be represented as:

$$\begin{aligned} \hat{f}_1^{\hat{J}} &= \underset{f_1^J, J}{\operatorname{argmax}} \left\{ Pr(f_1^J | c_1^K) \right\} \\ &= \underset{f_1^J, J}{\operatorname{argmax}} \left\{ Pr(c_1^K | f_1^J) \cdot Pr(f_1^J) \right\}, \end{aligned} \tag{3}$$

which suggests a decomposition into two sub-models:

1. Correspondence of the word sequence f_1^J and the character sequence c_1^K

   For one Chinese word sequence, its character sequence is unique. Hence, we can define the probability as one, if the character sequence of a word sequence is the same as the input, and as zero otherwise:

   $$Pr(c_1^K | f_1^J) = \begin{cases} 0 : C(f_1^J) \neq c_1^K \\ 1 : C(f_1^J) = c_1^K \end{cases}$$

   Here, C denotes the separation of a word sequence into characters.

2. The source language model at the word level:

$$\begin{aligned} Pr(f_1^J) &= \prod_{j=1}^{J} Pr(f_j | f_1^{j-1}) \\ &\cong \prod_{j=1}^{J} p(f_j | f_{j-n+1}^{j-1}) \end{aligned} \tag{4}$$

In practice, we use an n-gram language model as shown in the Equation 4.

Single-best segmentation

In the conventional approach, only the best segmentation $\hat{f}_1^{\hat{J}}$ is translated into the target sentence:

$$\hat{e}_1^{\hat{I}} = \underset{e_1^I, I}{\operatorname{argmax}} \left\{ Pr(e_1^I | \hat{f}_1^{\hat{J}}) \right\} \tag{5}$$

Segmentation lattice

In the transfer of the single-best segmentation from Equation 3 to Equation 5, some segmentations which are potentially optimal for the translation may be lost. Therefore, we combine the two steps. The search is then rewritten as:

$$\begin{aligned} \hat{e}_1^{\hat{I}} &= \underset{I, e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I | c_1^K) \right\} \\ &= \underset{I, e_1^I}{\operatorname{argmax}} \left\{ \sum_{f_1^J} Pr(f_1^J, e_1^I | c_1^K) \right\} \\ &= \underset{I, e_1^I}{\operatorname{argmax}} \left\{ \sum_{f_1^J} Pr(f_1^J | c_1^K) \cdot Pr(e_1^I | f_1^J, c_1^K) \right\} \\ &\cong \underset{I, e_1^I}{\operatorname{argmax}} \left\{ \max_{f_1^J} \left\{ Pr(f_1^J | c_1^K) \cdot Pr(e_1^I | f_1^J) \right\} \right\} \end{aligned} \tag{6}$$

Because our translation model in Equation 1 is based on the words, here we make the approximation that the target sentence e_1^I depends only on the word based source sentence f_1^J, but not on the character based one c_1^K. We also use the maximum instead of the sum over the segmentations.

In this way, the segmentation model and the translation model are combined into a model for the global decision.

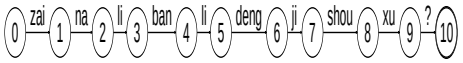

Figure 2: Segmentation lattice: input sentence at the character level as a linear automaton.

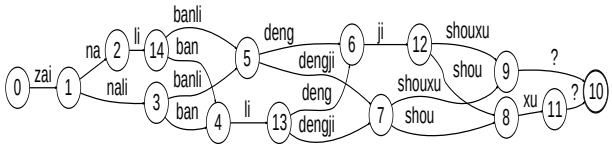

Figure 3: Segmentation lattice without weights.

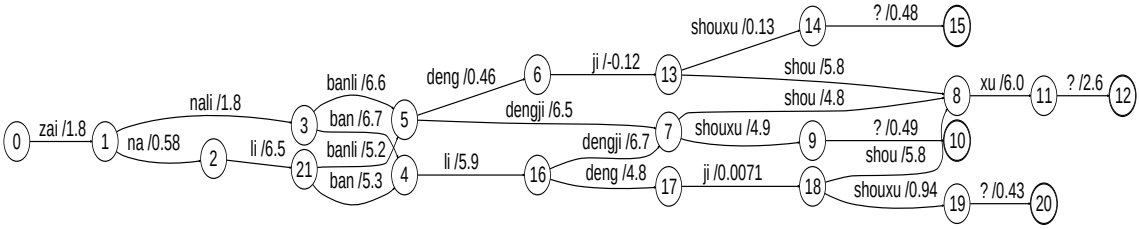

Figure 4: Segmentation lattice with language model weights.

3.4. Computational steps

Now we will take a short sentence as an example and simulate the segmentation process. The Chinese sentence is selected from the development corpus CStar'03 of the IWSLT 2005 task [4], its Pin-yin form is written in Table 1. The sentence consists of eight characters, including a punctuation mark. After the manual segmentation, it contains six words.

- Single-best segmentation

  Only the manually segmented sentence is translated. In this case, if any of the six words does not appear in the training corpus, its translation would be missing.

- Segmentation lattice

  The input sentence is at the character level as mentioned before. We generate the segmentation lattice with the following steps:

  1. We make a word list from the vocabulary of the manually segmented Chinese training corpus. Each word in the list is mapped by its characters as shown in Figure 2.

     To avoid the problem of the unknown characters from the unsegmented corpus, the additional characters from the test corpus are also added in the word list.

  2. We convert the mapping in Table 2 into a finite-state transducer for segmentation, as shown in Figure 5. Here the input labels are the characters from the test corpus, and the output labels will be concatenated with the Chinese training words in the translation system. The epsilon word is denoted as "eps", and the state 0 is the start and end state.

  3. Inside the translation systems, the input character sequence is represented as a linear acceptor, as shown in Figure 2.

  4. The linear automaton in Figure 2 is composed with the segmentation transducer in Figure 5. The result is a lattice which represents all possible segmentations of this sentence, as shown in Figure 3.

     Note that the alphabet in Figure 2 is a subset of the input alphabet in Figure 5, because the unknown characters are added to the word list as single words.

  5. With these steps, we get a new finite-state acceptor representing all the alternatives of different word segmentations. To have an integrated word segmentation in the translation, we only need to read segmentation lattice in Figure 3 instead of the manual segmented sentence.

Table 2: Word mapping from characters

Characters	Words
zai	zai
..	..
na li	nali
ban li	banli
deng ji	dengji
shou xu	shouxu

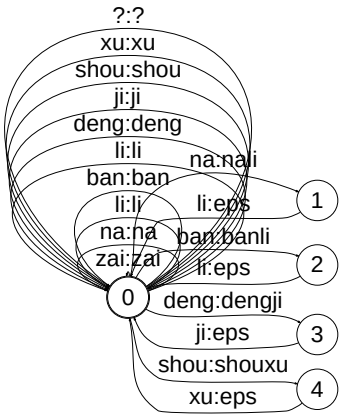

Figure 5: Segmentation transducer.

3.5. Weighting with language model costs

A problem of translation with the lattice in Figure 3 is that shorter paths are usually preferred because the search algorithm during translation finds the path with the smallest translation costs.

Therefore, we add word segmentation costs to a lattice. A word segmentation model represents the fluency of a Chinese word sequence and can be built as an n-gram language model of the word-based text. We trained the language model on the Chinese training corpus with the SRILM toolkit [7] and used the modified Kneser-Ney discounting.

To combine the segmentation lattice and the word based language model, we simply transform the language model into a finite-state transducer and compose the lattice with it.

After inserting the weights the number of nodes and arcs in a lattice may increase because of the language model histories.

4. Translation experiments

4.1. Task and corpus statistics

The translation experiments were carried out on the *Basic Travel Expression Corpus* (BTEC), a multilingual speech corpus which contains tourism-related sentences usually found in travel phrase books. We tested our system on the Chinese-to-English Supplied Task. The corpus was provided during the International Workshop on Spoken Language Translation [4]. The corpus statistics for the BTEC corpus are given in Table 3.

We used 19851 sentence pairs instead of 20000 due to corpus filtering. The Chinese texts in words are segmented manually. The evaluation data is the CStar'03 data set, whose Chinese text in words is the input to the single-best segmentation and the text in characters is the input to the segmentation lattice.

4.2. Evaluation criteria

So far, in machine translation research, a single generally accepted criterion for the evaluation of the experimental results does not exist. Therefore, we used different criteria: WER (word error rate), PER (position-independent word error rate), BLEU [12] and NIST [13]. For the evaluation corpus, we have sixteen references available. The four criteria are computed with respect to multiple references. The evaluation was case-insensitive. The BLEU and NIST scores measure accuracy, i.e. larger scores are better.

4.3. Evaluation results

We present the translation results on the IWSLT 2005 task [4] described in Section 4.1.

The experiments are based on two translation systems:

- Finite-state transducer-based translation

  In the finite-state transducer-based system we only use a monotone search because of the technical limitations of reordering with lattice input. Table 4 shows the results of the finite-state transducer based translations. Here, the translation using the single-best segmentation with a manually segmented input text has a BLEU score of 28.5%. By using the integrated segmentation, the BLEU score is increased by 0.5% absolute, and the NIST score by about 25% relative.

- Phrase-based translation

  The baseline results with the phrase-based translation have higher precision but also higher error rates as the results with the finite-state based translation. The reason is that many sentences translated by the finite-state transducer system are very short. There are only 2321 words in the translation hypothesis instead of 2521 words on average in the references. The phrase-based translation covered this shortcoming by including more feature functions as described in 2.3, especially the word penalty which can penalize shorter sentences.

  The baseline translation results of the phrase-based translation system have a BLEU score of 38.9%, as shown in Table 5. In our experiments, the reordering was taken at the phrase level and the model scaling factors were optimized on the evaluation data with respect to the combination of all the criteria. Here, using the segmentation lattice with a bi-gram source language model, the improvement in the BLEU score is 1.5% absolute compared to the baseline, and the WER and PER are reduced by 11.9% and 13.2%, respectively.

4.4. Computational Requirement

We use the lattice density to measure the size of a segmentation lattice, which is defined as the number of arcs in the lattice divided by the number of characters in the sentence.

Table 3: Corpus statistics

		Chinese		English
Train:	Sentences	19 851		
	Running Words	18 1247		159 655
	Vocabulary	7 610		6 955
	Singletons	3 512		2 938
CStar'03:	Sentences	506		
		Words	Characters	Words
	Running Words/Characters	3 515	4 757	65 604
	Vocabulary	870	800	2 078
	OOVs (running words/characters) [%]	5.40	8.74	14.3
	OOVs (in vocabulary) [%]	18.4	26.3	20.6

Table 4: Translation performance with monotone finite-state transducer based translation for different segmentation methods.

Segmentation methods	WER [%]	PER [%]	NIST	BLEU [%]
Single-best (manual) segmentation	51.3	43.1	3.60	28.5
Segmentation lattice without weights	51.6	42.2	4.69	29.0

Table 5: Translation performance with phrase-based translation for different segmentation methods.

Segmentation methods	WER [%]	PER[%]	NIST	BLEU[%]
Single-best (manual) segmentation	53.6	43.8	8.18	38.9
Segmentation lattice without weight	47.0	38.1	8.09	40.2
Segmentation lattice with bi-gram LM	47.2	38.0	8.18	40.4

For the 506 sentences in the evaluation set, on average, the density of the lattices without weights is 1.5, and it is 3.9 with bi-gram language model weights.

The memory requirements with different segmentation methods for translation of the CStar'03 data set are as following: with the single-best segmentation, it is 54.2 MB, and with the segmentation lattice not using a source language model, it is 56.9 MB. If we use a bi-gram source language model the requirement increases to 65.8 MB.

The translation speed using the segmentation with lattice is 0.266 second per sentence, it is almost as fast as the translation using the single-best segmentation, i.e. 0.262 second per sentence. By using a bi-gram source language model, the speed slows down to 0.820 second per sentence.

5. Discussion and future work

We have successfully developed a new Chinese word segmentation method for statistical machine translation. The method combines the segmentation decisions directly in the search for the translations, which has two major advantages:

1. The Chinese input text is on character level. There is no need to segment the text during pre-processing.
2. The translation system with the integrated segmentation outperforms the one that uses single-best (manual) segmentation.

In the experiments on the IWSLT task 2005 [4], the integrated segmentation approach outperforms the single-best segmentation using both the finite-state transducer based and phrase-based systems. With the phrase-based system, the BLEU score is increased by 1.5% absolute. Although these are promising results, so far the changes in word segmentation are only carried out in the translation process. As we mentioned in Section 3.1, to minimize the number of lexicon entries, we can try to perform a better segmentation in training. [14] suggested a way to perform the phrase segmentation and alignment in one step.

By refining our model, we expect a further improvement with the integrated word segmentation method.

6. Acknowledgments

This work was partly funded by the DFG (Deutsche Forschungsgemeinschaft) under the grant NE572/5-1, project "Statistische Textübersetzung" and the European Union under the integrated project TC-Star (Technology and

Corpora for Speech to Speech Translation, IST-2002-FP6-506738, http://www.tc-star.org).

7. References

[1] J. Xu, R. Zens, and H. Ney, "Do we need Chinese word segmentation for statistical machine translation?" in *Proc. of the Third SIGHAN Workshop on Chinese Language Learning*, Barcelona, Spain, July 2004, pp. 122–128.

[2] H. Ney, "Speech translation: Coupling of recognition and translation," in *Proc. of IEEE Intl. Conference on Acoustics, Speech and Signal Processing*, Phoenix, AZ, March 1999, pp. 1149–1152.

[3] X. Luo and S. Roukos, "An iterative algorithm to build Chinese language models," in *Proc. of the 34th annual meeting of the Associaton for Computational Linguistics*, Santa Cruz, California, June 1996, pp. 139–143.

[4] IWSLT, "Intl. workshop on spoken language translation home page," 2005, http://www.is.cs.cmu.edu/iwslt2005/CFP.html.

[5] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "A statistical approach to machine translation," *Computational Linguistics*, vol. 16, no. 2, pp. 79–85, June 1990.

[6] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, March 2003.

[7] A. Stolcke, "SRILM - an extensible language modeling toolkit." in *Proc. of Intl. Conference on Spoken Language Processing*, Denver, Colorado, September 2002, pp. 901–904.

[8] S. Kanthak and H. Ney, "FSA: An efficient and flexible C++ toolkit for finite state automata using on-demand computation," in *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, July 2004, pp. 510–517.

[9] F. Casacuberta, D. Llorens, C. Martinez, S. Molau, F. Nevado, H. Ney, M. Pasto, D. Pico, A. Sanchis, E. Vilar, and J. Vilar, "Speech-to-speech translation based on finite-state transducer," in *Proc. of IEEE Intl. Conference on Acoustics, Speech and Signal Processing*, Salt Lake City, Utah, May 2001, pp. 613–616.

[10] R. Zens and H. Ney, "Improvements in phrase-based statistical machine translation," in *Proc. of the Human Language Technology Conference*, Boston, MA, May 2004.

[11] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July 2003, pp. 160–167.

[12] K. A. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, July 2002, pp. 311–318.

[13] G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *Proc. of Human Language Technology*, San Diego, California, March 2002, pp. 128–132.

[14] Y. Zhang, S. Vogel, and A. Waibel, "Integrated phrase segmentation and alignment algorithm for statistical machine translation," in *Proc. of Intl. Conference on Natural Language Processing and Knowledge Engineering (NLP-KE'01)*, Beijing, China, October 2003, pp. 567–573.