Proceedings of the Workshop on Speech-to-Speech Translation:
Algorithms and Systems, Philadelphia, July 2002, pp. 23-30.
Association for Computational Linguistics.

Finding Translation Pairs from English-Japanese Untokenized Aligned Corpora

Genichiro KIKUI and Hirofumi YAMAMOTO
ATR Spoken Language Translation Research Laboratories
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0288, JAPAN

Abstract

This paper proposes a new algorithm for finding translation pairs in an English-Japanese parallel aligned corpus. Unlike previous methods, our algorithm does not presuppose a separate tokenizer for Japanese, but finds translation pairs as "side-effects" of unsupervised tokenization of Japanese sentences by using information from the English sentences. The algorithm is based on the observation that two Japanese sentences tend to have a common word when their English mates (i.e., aligned sentences) contain the same word. We implemented this idea as an unsupervised tokenization of Japanese with extended Hidden-Markov-Models (HMMs), where hidden n-gram probabilities (i.e., state transition probabilities) are affected by co-occurring words in the English part. Our experiment on finding noun-noun translation pairs achieved 76.3% accuracy, which was 0.4 points lower than the result using supervised tokenization.

1 Introduction

As pointed out by Fung (Fung, 1995), automatic compilation of a domain-specific bilingual lexicon from corpora is needed to improve machine translation, cross-language information retrieval, etc.

Actually, a number of methods have been already proposed for finding translation pairs from aligned sentences (Dagan et al., 1993), (Wu and Xia, 1994). These methods presuppose sentences from both languages are segmented (or tokenized) into words (e.g.,(Wu and Xia, 1994)). Thus, when applied to a language whose written form does not have trivial word boundaries (e.g., Chinese, Japanese, and Korean), sentences have to be segmented first by a separate tokenizer. Although accurate tokenizers have been proposed for a limited number of languages, they depend on the rules of individual languages or require large segmented corpora for training. Thus, they cannot be easily transferred to other languages. Another problem of relying on a separate tokenizer is that word boundaries determined by the tokenizer are not always optimal for constructing a bilingual lexicon.

This paper proposes an algorithm for finding English-Japanese translation pairs from aligned corpora while simultaneously tokenizing the Japanese part of the corpora. The underlying idea of our algorithm is that two Japanese sentences tend to have a common word when their English mates (i.e., aligned sentences) contain the same word. We implemented this idea as an unsupervised tokenization of Japanese using an extended version of Hidden-Markov-Models (HMMs), where n-gram probabilities (i.e., state transition probabilities) are affected by co-occurring words

in the English sentences.

Since the algorithm does not rely on a specific feature of Japanese[1] or trained tokenizers, it is applicable to parallel corpora consisting of English and another language without trivial word boundaries. We expect that this kind of corpora will be available for many languages in the near future because texts in a local language will be translated into English and vice versa, considering that English is the "common" language of the world.

The remaining part of the paper is organized as follows. Section 2 introduces the problem. Section 3 presents our algorithm. Section 4 reports the results of the experiments and the discussion.

2 Problem Setting

2.1 Aligned Corpus

Input to our algorithm is an English-Japanese parallel aligned corpus, where every sentence in English is paired with its translation in Japanese. An example of pairs is shown in Figure 1. Although this paper uses an English-Japanese corpus, the proposed algorithm is applicable to any language pair, one from a language with trivial word boundaries and the other from a language without them.

2.2 Finding Translation Pairs from Untokenized Corpus

Our goal is to find translation pairs from the parallel corpus described above. This goal is achieved by performing the following two:

1. Identifying words of aligned sentences, what we call *segmentation* or *tokenization* and

2. Finding translation pairs from the segmented corpora, what we call *extraction*.

Existing approaches, most of which are for pairs of Western-European languages, concentrate on the second task, since words in these languages are already separated by explicit markers, such as spaces. For East-Asian languages, (e.g., Chinese, Japanese, and Korean), however, the first task is equally important, since these languages have no explicit word boundaries in their written form.

Let us consider a hypothetical language, L_X, instead of Japanese. L_X has 10 words, as shown in Figure 2, and its written form does not have explicit word separators.

Alphabet	A, B, C, D, E, F
Lexicon	A, B, BC, CDE, CEF, DE, DF, EF, F, FB

Figure 2: Hypothetical language L_X.

Then, suppose our goal is to find English-L2 translation pairs, e.g., "paper-BC", from a parallel corpus shown in Figure 3.

English	L_X
E01: Here is a paper.	L01:ABCDE
E02: He wrote a paper.	L02:FBCEF
E03: He read the paper.	L03:FBCDF

Figure 3: Hypothetical aligned corpus.

Since we presuppose that no external information is available besides the lexicon, a sentence in L_X can be segmented in many ways, as shown in Figure 4.

	possible segmentations
L01	A/B/CDE, A/BC/DE
L02	FB/CEF, F/BC/EF, F/B/CEF
L03	FB/C/DF, F/BC/DF

Figure 4: Possible Segmentations for L01 - L03. ("/" shows a word boundary.)

The problem tackled in this paper is how to find a translation of each English noun from possible words in the L_X, or Japanese, side of the parallel corpus.

3 Algorithm

Our algorithm is an extension of unsupervised segmentation using Hidden-Markov-Models (HMMs) (Takeuchi and Matsumoto, 1995). The original segmentation algorithm tries to estimate HMM parameters in such a way that

[1] e.g., character classes

English	Japanese
Please show me your passport.	パスポートを見せてください
Tea or coffee?	コーヒーと紅茶、どちらになさいますか

Figure 1: Sample Aligned Corpus

the total entropy of the (unsegmented) training corpus is maximized.

This approach has a great advantage since it is free from language dependent heuristics and since in such applications as speech recognition, the estimated parameters sometimes achieve even better performance than those estimated by supervised training. The problem, however, is that it cannot definitely guarantee that the resulting word boundaries are semantically correct.

In order to solve this problem, our segmentation algorithm tries to include semantic information by using the aligned English sentences. More specifically, we extended the original unsupervised segmentation algorithm based on the following assumption:

> If a Japanese word in a segmentation candidate has its translated word in the aligned English sentence, the Japanese word gets higher probability according to the strength of their translation relation.

We have no bilingual dictionary, which is the output of our algorithm, so the translation relation between two words should be calculated somehow from the parallel corpus, e.g., by using mutual information. This calculation, however, requires solving the initial problem of segmenting Japanese sentences.

To get around this chicken-and-egg problem, our algorithm iteratively estimates translation relations, beginning with tentatively segmenting Japanese sentences without using bilingual information, then estimating translation relations using the tentative segmentation. Next, the algorithm re-segments Japanese sentences considering the translation relations, resulting in updated translation relations.

In order to cope with ambiguous segmentation, our algorithm estimates translation relations using every word in possible segmentations by assuming that each word's occurrence count is not a natural number but the probability of the word calculated by the HMM segmentation model.

This process is equivalent to relating an English word and Japanese word if they co-occur in many translation pairs.

For example, the sample L_X corpus shown in Figure 3 includes three English sentences that share the word "paper". The word "BC" is a candidate for the translation of "paper" since it appears in the set of candidates for every sentence in L_X (as shown in 4).

The remaining part of this section first explains unsupervised learning of an n-gram model, then extends it for handling a parallel corpus.

3.1 Unsupervised Sentence Tokenization using Markov Models

Let us concentrate on tokenizing a Japanese monolingual sentence.

In the statistical framework, the most likely word-sequence $\hat{W}$ for a sentence S is formalized as follows (Takeuchi and Matsumoto, 1995), (Nagata, 1994):

$$\hat{W} = \arg\max_{W} P(W), \qquad (1)$$

where $P(W)$ is the probability of a word sequence $W = w_1, \ldots, w_n$ whose surface string is equal to that of S. This formula is transformed to the following form, known as the *n-gram model*, by approximating the probability of each word depending on N preceding words.

$$\hat{W} = \arg\max_{W} \prod_{1}^{|W|} P(w_i | w_1, \ldots, w_{i-1})$$

$$= \arg\max_{W} \prod_{1}^{|W|} P(w_i | w_{i-N+1}, \ldots, w_{i-1}) \quad (2)$$

A larger N will make the model stronger, but a larger amount of data is required for estimating probabilities. Thus, a value of 2 or 3 is usually used. Hereafter, we fix N to 2 for simplicity.

In order to implement a tokenizer using the n-gram model, we must solve the following problems.

1. How to efficiently find $\hat{W}$ for given n-gram probabilities $P(w_i \mid w_{i-1})$.
2. How to estimate n-gram probabilities.

Since the first problem can be solved by using the dynamic programming technique, we concentrate on the second problem.

The standard way to answer the second question is to find the probability distributions that can generate the given corpus with highest probability. Formally, this is achieved by finding the probability distributions that maximize the log-likelihood $log(L)$ of the training corpus. If the training corpus is already segmented, $log(L)$ is easily calculated as shown:

$$log(L) = \sum_{w_i \in Corpus} log P(w_i | h_i), \quad (3)$$

where h_i is the history of the word, w_i. In the bigram case, h_i is simply the preceding word, w_i.

In our problem, however, the corpus has no explicit segmentation. Thus, we consider every possible segmentation for each string as follows:

$$log(L) = \sum_{w_{p,s} \in Corpus} log P(w_{p,s} | +), \quad (4)$$

where $w_{p,s}$ corresponds to a word whose surface form is s, starting from the position p, and $+$ represents the context of $w_{p,s}$, in this case, the previous word.

The probability distributions P that maximize the above formula cannot be determined analytically but by an iterative algorithm called the *Forward-Backward Algorithm* or the *Baum-Welch Algorithm* as roughly described below [2].

Note that the algorithm presupposes possible segmentations and $P(w_{p,s}|h)$ are represented by a lattice.

- Step 0 (Initialization)

  Assign $P(w_{p,s} \mid +)$ to every word in the lattice.

  Since $P(w_{p,s} \mid +)$ is unknown, we use the 0-gram (uniform) probability as the initial value.

- Step 1

  For each sentence, estimate posterior (bigram) probability for each word in each (Japanese) sentence J_k, written as $P(w_{p,s}, +|J_k)$.

  Posterior probabilities are calculated by using *forward and backward probabilities* (Jurafsky and Martin, 2000), (Manning and Schuetze, 1999).

- Step 2

  Re-estimate the bigram (prior) probabilities based on the maximum likelihood estimation, regarding the posterior probability of each word as the real-valued occurrence frequency. Formally,

$$P(w_{p,s} \mid +) = \frac{\sum P(w_{p,s}, +|J_k)}{\sum P(+|J_k)}. \quad (5)$$

- Step 3

  If some condition is satisfied, then stop, otherwise go to step 2. The condition could be the iteration count or total entropy.

[2] Detailed explanations are found in textbooks (Rabiner and Juang, 1993), (Jurafsky and Martin, 2000), (Manning and Schuetze, 1999)

3.2 Including Aligned Sentences

Since the algorithm in the previous section tries to segment each sentence to maximize the log-likelihood, it is not necessarily optimal for finding translation pairs.

In this section, we modify the algorithm under the assumption presented in Section 2.1. We begin by considering that the probability of a sentence in Japanese, J_k, is conditioned by the aligned sentence, E_k. Thus, the log-likelihood of this model is

$$log(L) = \sum_{(J_k,E_k)\in Corpus} logP(J_k|E_k). \quad (6)$$

We can rewrite this formula in such a way that each Japanese word depends on both the preceding word(s) and the aligned sentence.

$$P(J_k \mid E_k) = \prod_{w_{p,s}\in J_k} P(w_{p,s} \mid +, E_k) \quad (7)$$

It is, however, hard to estimate a reliable value for $P(w_{p,s} \mid +, E_k)$ since it includes the entire sentence, E_k. One possibility is to use Maximum Entropy Markov Models (MEMM) (McCallum et al., 2000). In this paper, we take a simpler approach.

First, we decompose the above conditional probability into a monolingual part and a bilingual part.

$$P(w_{p,s} \mid +, E_k) = P(w_{p,s} \mid +)P(w_{p,s} \mid E_k). \quad (8)$$

The monolingual part is exactly same as the previous n-gram probability. The bilingual part, $P(w_{p,s} \mid E_k)$, is expressed as the sum of all possible alignment A (Brown et al., 1990):

$$P(w_j|E_k) = \sum_A P(w_j, A|E_k). \quad (9)$$

Since A just specifies $a(w_j)$, i.e., the translation of w_j, the above formula can be approximated as follows:

$$\begin{aligned} P(w_j|E_k) &= \\ &\sum_A P(w_j|a(w_j))P(A|w_j, E_k) \\ &= \sum_A P(w_j|a(w_j))P(a(w_j)|w_j). \end{aligned} \quad (10)$$

Instead of summing up every A, which means to consider all the words in E_k, we approximate the formula by using the maximum value [3]

$$P(w_j|E_k) \simeq \hat{P}(w_j|E_k) = \max_{w_e\in E_k} P(w_j \mid w_e)P(w_e \mid w_j) \quad (11)$$

We estimate $P(w_j \mid w_e)P(w_e \mid w_j)$ by counting numbers of co-occurring words:

$$P(w_j \mid w_e)P(w_e \mid w_j) = \frac{C^2(w_j, w_e)}{C_j(w_j)C_e(w_e)}, \quad (12)$$

where $C(w_j, w_e)$ is the number of times w_j and w_e co-occur in one aligned pair, $C_j(w_j)$ is the frequency of w_j in the Japanese part of the aligned corpus and $C_e(w_e)$ is the frequency of w_e in the English part.

To estimate $P(w_i \mid +, E_k)$, we use the algorithm shown in Section 3.1 replacing $P(w_i \mid +)$ with $P(w_i \mid +, E_k)$.

1. Step 0 (same as in Section 3.1)
2. Step 1 (same as in Section 3.1)
3. Step 2

   Re-estimate $P(w_{p,s} \mid +, S_e)$ in the following form:

   $$P(w_{p,s} \mid +, S_e) = P(w_{p,s} \mid +)P(w_{p,s}|E_k), \quad (13)$$

   where $P(w_{p,s} \mid +)$ is given by (5) and $P(w_{p,s}|E_k)$ is given by (11). Note that $C_j(w_j)$ is calculated by summing the posterior probabilities of w_j in the entire Japanese segmentation candidates.
4. Step 3 (same as in Section 3.1)

[3] This roughly corresponds to using the Viterbi Alignment instead of all the possible alignments.

3.3 Bilingual Dictionary Construction

After segmentation is completed, translation pairs can be found by simply retrieving such combinations of w and e where $P(w_j, w_e)$ and/or $C(w_j, w_e)$ are over a certain threshold.

4 Evaluation Experiments

We applied the proposed algorithm to a Japanese-English corpus of travel expressions. The corpus includes 200 k sentence pairs that are considered to be frequently used in traveling abroad. This corpus is divided into two non-overlapping sets: a *training set* with 190 k sentences and a *test set* with 0.5 k sentences [4].

The lexicon of Japanese is compiled by accumulating all the entries of the dictionary of ChaSen (Matsumoto et al., 1997), a Japanese morphological analyzer. Every word in the English part of the corpus was assigned a part-of-speech. Using the part-of-speech, we restricted ourselves to finding translations of nouns.

In the following experiments, we modified the formula (8) as:

$$P(w_{p,s} \mid +, S_e) = (1-\lambda)P(w_{p,s} \mid +) + \lambda P(w_{p,s}|E_k) \quad (14)$$

where λ determines the weight of the translation probability.

4.1 Segmentation Accuracy

First, we evaluated the segmentation performance in terms of the entropy, word accuracy and noun recall for the test corpus. The relation between total entropy and the iteration count is shown in Figure 5.

The *word accuracy* is defined as follows:

$$acc = \frac{Ins + Del + Sub}{TotalWordCount}.$$

where Ins, Del, and Sub are, respectively, counts of insertion, deletion, and substitution words as compared with the reference data.

[4] The remaining sentences are reserved.

The *noun recall* is how many nouns in the correctly segmented sentences are identified as words (i.e., nouns) by the tokenizer.

$$\frac{\#\,of\ nouns\ correctly\ identified\ by\ the\ system}{total\ \#\,of\ nouns}$$

The values for different λ are shown in Table 1. For comparison, results of supervised tokenization (Nagata, 1994)[5] were included in the table.

Since the unsupervised algorithm tries to find segmentation such that the total entropy is minimized[6], the total entropy for each result is smaller than that of supervised learning. This shows that our method can produce segmentations useful for building language models for speech recognition etc.

On the other hand, word accuracies, which are roughly linear to segmentation accuracies, are worse than the results of supervised tokenization, as expected. Many errors were found in functional words and suffixes. Functional words are often wrongly concatenated to adjacent functional words when they occur frequently. The unsupervised tokenizer often separates an inflexive suffix of a verb from the main part, although the reference segmentation treats them as one unit.

The noun recall for each run is significantly greater than the word accuracy. This shows that translation relations are useful for obtaining correct word boundaries and identifying nouns.

Considering the fact that the number of functional words is limited and a small training corpus improves segmentation accuracy(Takeuchi and Matsumoto, 1995), the best method would be to combine the supervised method with our algorithm which is effective for content words.

4.2 Correctness of Translation Pairs

Next, we investigated how correctly our algorithm could compile a bilingual dictionary. In this evaluation, we collected every bilingual pair, j, e, that satisfied the following condition:

[5] We ignored part-of-speech .

[6] More accurately, "locally minimized".

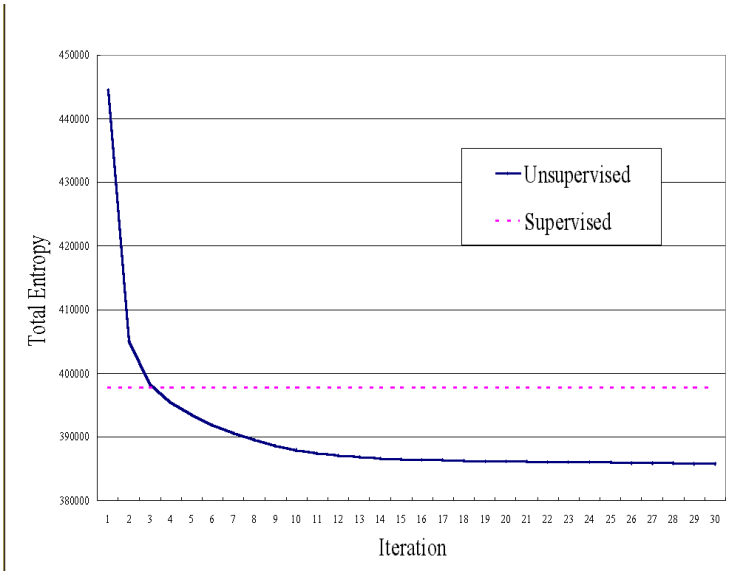

Figure 5: Entropy vs. Iteration Count

Table 1: Word Accuracy

λ	total entropy	word accuracy	noun recall
0.0	385801.85	68.0	86.3
0.25	386569.83	70.2	87.8
0.5	386983.16	70.5	89.2
0.75	388118.42	71.9	89.7
0.90	388751.05	72.2	90.1
supervised	397785.54	98.7	98.1

$C(w_j, w_e) \geq M (M = 1, \ldots, 5)$.

This condition states that the co-occurrence frequency of the translation pair should be greater than or equal to a threshold, M.

We compared our algorithm with the following two settings.

1. Dictionary Only (DIC)

   As a baseline, we stopped our procedure after the first iteration. This means that every possible segmentation is equally likely.

2. Supervised Segmentation (SUP)

   This case also stops the procedure before re-estimation, but instead of uniform distribution for initialization, we used probabilities estimated from a manually segmented training corpus.

The correctness of every translation pair produced by the above methods was evaluated by professional translators. The results are shown in Table 2, where accuracy is defined as:

$$\frac{\#\ of\ correct pairs}{total\ \#\ of\ outputs}$$

This table shows that our algorithm is more accurate and found more bilingual pairs than the method using every possible word and is slightly less accurate than the method using supervised segmentation.

5 Concluding Remarks

In this paper, we proposed an algorithm that extracts translation pairs from an English-Japanese parallel raw corpus. The unique feature of our algorithm is that it does not

Table 2: Accuracy of Translation Pairs
DICT = dictionary only, UNS = proposed, SUP = supervised

M (min. occ.)	correct DIC/UNS/SUP	partially correct DIC/UNS/SUP	wrong DIC/UNS/SUP	# found DIC/UNS/SUP
5	74.5/76.3/76.7	14.5/15.5/16.4	11.1/ 8.2/ 6.9	1545/1788/1837
4	73.8/75.4/76.2	15.0/15.9/16.8	11.2/ 8.7/ 7.0	1769/2012/2068
3	73.1/73.4/75.0	15.4/17.3/17.7	12.0/ 9.3/ 7.3	2072/2398/2423
2	66.9/69.0/70.9	17.2/19.0/18.8	15.9/12.0/10.3	2709/3174/3240
1	52.0/57.0/57.5	17.9/20.0/20.3	30.1/23.0/22.2	5431/5860/5884

require any separate tokenizer for Japanese. This means that the algorithm can find translation pairs from any parallel corpus consisting of a language with trivial word boundaries and a language without them. Experimental results showed that the algorithm achieved only a 0.4 points lower accuracy than supervised segmentation.

The proposed algorithm can be elaborated in many ways. One direction would be to use a better formula for the degree of translation relation. Another direction would be to improve the method of combining an n-gram probability and a translation probability. The maximum entropy approach as stated before (McCallum et al., 2000) for coupling two probabilities in a principled way is promising.

6 Acknowledgment

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

P. Brown, J. Cocke, V. Della Pietra, F. Jelinek, R.L. Mercer, and P. C. Roosin. 1990. A statistical approach to language translation. *Computational Linguistics*, 16(2):79–85.

Ido Dagan, Kenneth Church, and William Gale. 1993. Robust word alignment for machine aided translation. In *Proceedings of the Workshop on Very Large Corpora*, pages 1–8.

Pascale Fung. 1995. Compiling bilingual lexicon entries from a non-parallel English-Chinese corpus. In David Yarovsky and Kenneth Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 173–183, Somerset, New Jersey. Association for Computational Linguistics.

Daniel S. Jurafsky and James H. Martin. 2000. *SPEECH and LANGUAGE PROCESSING.* Prentice Hall.

Christoper D. Manning and Hinrich Schuetze. 1999. *Foundations of Statistical Natural Language Processing.* The MIT Press.

Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, O. Imaichi, and T. Imamura. 1997. *Japanese morphological analysis system ChaSen manual.* Technical Report NAIST-IS-TR97007, Nara Institute of Technology.

Andrew McCallum, Dayne Freitag, and Fernando Pereira. 2000. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of 17th International Conf. on Machine Learning*, pages 591–598. Morgan Kaufmann, San Francisco, CA.

Masaaki Nagata. 1994. A stochastic Japanese morphological analyzer using a forward-dp backward-a* n-best search algorithm. In *Proceedings of COLING-94*, pages 201–207.

Lawrence Rabiner and Biing-Hwang Juang. 1993. *Foundations of Speech Recognition.* Prentice Hall.

Kouichi Takeuchi and Yuji Matsumoto. 1995. HMM parameter learning for Japanese morphological analyzer. In *Proceedings of PACLING95*, pages 163–172.

Dekai Wu and Xuanyin Xia. 1994. Learning an English-Chinese lexicon from a parallel corpus. In *Proceedings of the First Conference of the Association for Machine Translation in the Americas (AMTA-94).*