HLT-NAACL 2003 Workshop: Building and Using Parallel Texts
Data Driven Machine Translation and Beyond , pp. 104-110
Edmonton, May-June 2003

Input Sentence Splitting and Translating

Takao Doi, Eiichiro Sumita
ATR Spoken Language Translation Research Laboratories
2-2-2 Hikaridai, Kansai Science City, Kyoto, 619-0288 Japan
{takao.doi, eiichiro.sumita}@atr.co.jp

Abstract

We propose a method to *split and translate* input sentences for speech translation in order to overcome the long sentence problem. This approach is based on three criteria used to judge the goodness of translation results. The criteria utilize the output of an MT system only and assumes neither a particular language nor a particular MT approach. In an experiment with an EBMT system, in which prior methods cannot work or work badly, the proposed split-and-translate method achieves much better results in translation quality.

1 Introduction

To achieve translation technology that is adequate for speech translation, the possibilities of several corpus-based approaches are being investigated. Among these methods, DP-match Driven transDucer (D^3) has been proposed as an Example-Based Machine Translation (EBMT). When D^3 is adapted to Japanese-to-English translation in a travel conversation domain, the method can achieve a high translation quality (Sumita, 2001 and 2002). On the other hand, the translation method is sensitive to the long sentence problem, where longer input sentences make it more difficult for a machine translation (MT) system to perform good translation. To overcome this problem, the technique of splitting an input sentence [1] and translating the split sentences appears promising.

The methods of previous studies related to this approach can be roughly classified into two types: one splits sentences before translation and the other splits them in the parsing phase of translation. We'll call the former pre-process-splitting, the latter parse-time-splitting, and translation with any splitting split-and-translate.

In previous research on pre-process-splitting, such as Takezawa (1999), many methods have been based on word-sequence characteristics. Some research efforts have achieved high performance in recall and precision against correct splitting positions. Despite such a high performance, from the view point of translation, MT systems are not always able to translate the split sentences well.

In some research works on parse-time-splitting, such as Furuse (1998 and 2001), sentences have been split based on parsing trees under construction. Partly constructed trees are combined and translated. A sentences is split according to the sub-trees. The split sentences can be translated because an internal parsing mechanism guarantees their fitness. However, parse-time-splitting technique cannot be adapted, or can be adapted only as pre-process-splitting by using an external parsing system, to MT systems that deal with no parsing tree, such as D^3 and Statistical MT.

In this paper, we propose another split-and-translate technique in which splitting and translation act in harmony. This technique depends on no particular MT method, therefore can be applied to D^3. In order to prove the effect for translation quality, our proposed split-and-translate method and, for the purpose of comparison, a pre-process-splitting technique are evaluated. For convenience, we'll call the two split-and-translate methods in our experiments as follows.

method-T: Our proposed method based on partial *T*ranslation results, described in section 2.
method-N: Before translation, splitting an input sentence with the pre-process-splitting method based on *N*-gram, described in section 3.

The following sections describe the two methods, the MT system, D^3 that the methods are applied to, and experiments.

[1] Strictly speaking, this isn't necessarily a sentence but an utterance including sentences. In this paper, we use the term *sentence* without strictly defining it to simplify discussion.

2 Proposed Split-and-Translate: Method-T

An MT system sometimes fails to translate an input, for example, due to failure in parsing a sentence or retrieving examples. Such a failure occurs particularly when an input is longer. In such a case, by splitting the input, translation may be successfully performed for each portion. Therefore, one idea is to arrange the translations of split portions in the same order as in the source sentence and to consider the arrangement as a translation of the entire input sentence. Particularly in a dialogue, sentences tend not to have complicated nested structures, and many long sentences can be split into mutually independent portions. Therefore, if splitting positions and translations of split portions are adequate, the possibility that this simple arrangement of the translations can provide an adequate translation of the complete input is relatively high.

In the example below, a Japanese sentence (1-j) has potentially adequate splitting positions such as (1-j'). The arrangement of the English translations of the portions (1-e) is an adequate translation.

(1-j) sou desu ka ee kekkou desu jaa tsuin de o negai shi masu[2]
(1-j') sou desu ka | ee | kekkou desu | jaa | tsuin de o negai shi masu
(1-e) i see | yes | that's fine | then | a twin please

2.1 Criteria

When you split-and-translate a sentence, some portions can be translated while others cannot. We call the count of words in the portions that cannot be translated the *fault-length*. It is natural to consider (X) as a criterion to judge the goodness of split-and-translate results.

(X) The smaller the *fault-length* is, the better the result is.

Let the term partial-translation be the translation of a portion that can be translated. In a split-and-translate result, there can be some partial-translations. *Partial-translation-count* expresses the number of partial-translations. (Y) is also a natural criterion to judge the goodness of a split-and-translate result.

(Y) The smaller the *partial-translation-count* is, the better the result is.

Many current MT methods produce not only target sentences but also scores. The meaning of a score, depending on the translation method, can be parsing cost, distance between sentences, word correspondence probability, or other meanings or combinations of the above. If there is a correlation between the score and the translation quality, we can make use of this score as a confidence factor of translation. We can use the confidence factor as another criterion for split-and-translate results. In order to ensure reliability for the complete result of split-and-translate procedures from confidence factors, the scores of all partial-translations are combined. We call this combined score the *combined-reliability*. How to combine scores depends on the mathematical characteristics of the scores. Therefore the third criterion (Z) is added.

(Z) The higher the *combined-reliability* is, the better the result is.

From the above considerations, the proposed method utilizes these criteria to judge the goodness of split-and-translate results with the priority as follows.

1. The smaller the *fault-length* is, the better the result is.
2. Unless judged with criterion-1, the smaller the *partial-translation-count* is, the better the result is.
3. Unless judged with criterion-1 or criterion-2, the higher the *combined-reliability* is, the better the result is.

The case where translation can be performed without splitting meets these criteria. In this case, the *fault-length* is 0, the *partial-translation-count* is 1, and the *combined-reliability* equals the score of the complete translation that must be utilized by the MT system; therefore, this result is the best.

Criterion-3 has a low priority. Unless an MT system has a confidence factor, only criteria-1-2 are used.

These three criteria are based on the output of an MT system, that is, how well the MT system can translate portions. Split portions are translated, and the partial-translation results are evaluated to select the best split positions (the algorithm is discussed in section 6). As the proposed split-and-translate method is based on these criteria only, this method assumes no parsing process and depends on neither a particular language nor a particular MT method.

2.2 Example

Below, we show an example of selecting a result from candidates based on criteria-1-2.

(2-j) hai wakari mashi ta sore to ne choushoku na n desu

[2] Its English translation is "I see. Then, fine, we'll take a twin." in a corpus

kedomo dou nat teru n deshou ka[3]

(2-j') hai | wakari mashi ta | sore to ne | choushoku na n desu kedomo | dou nat teru n deshou ka

For a Japanese input (2-j), there are many candidates of splitting points such as (2-j'). We consider three splittings: (2-a), (2-b) and (2-c).

(2-a) hai wakari mashi ta | sore to ne choushoku na n desu kedomo dou nat teru n deshou ka
(2-b) hai wakari mashi ta sore to ne choushoku na n desu kedomo | dou nat teru n deshou ka
(2-c) hai wakari mashi ta | sore to ne choushoku na n desu kedomo | dou nat teru n deshou ka

Suppose the partial translations corresponding to these candidates are as follows, where *fault-lengths* and *partial-translation-counts* are calculated.

(2-a')
hai wakari mashi ta => yes i see
sore to ne cyoushoku na n desu kedomo dou nat teru n deshou ka
=> and what about breakfast
fault-length = 0
partial-translation-count = 2
(2-b')
hai wakari mashi ta sore to ne choushoku na n desu kedomo => FAIL
dou nat teru n deshou ka => what happened to it
fault-length = 12
partial-translation-count = 1
(2-c')
hai wakari mashi ta => yes i see
sore to ne choushoku na n desu kedomo => and i breakfast
dou nat teru n deshou ka => what happened to it
fault-length = 0
partial-translation-count = 3

(2-a) and (2-c) are better than (2-b) based on criterion-1, and (2-a) is better than (2-c) based on criterion-2, so the rank is (2-a), (2-c), (2-b).

3 Pre-Process-Splitting for Method-N

For splitting input sentences as a pre-process of MT systems, we consider a previous study of pre-process-splitting. Many pre-process-splitting methods are based on word-sequence characteristics. Among them, we use the method of Takezawa (1999), a pre-process-splitting based on the N-gram of part-of-speech subcategories. This method is derived from that of Lavie (1996) and modified especially for Japanese.

The function of this method is to infer where splitting positions are. Splitting positions are defined as positions at which we can put periods. For each position, to calculate the plausibility that the position is a splitting position, we consider the previous two words and the following one word, three words in total. Part-of-speech and conjugation-type are considered as word characteristics. When the plausibility is higher than a given threshold, the position is regarded as a splitting position. The threshold is manually selected to tune the performance for a training set. Equation [1] shows how to calculate the plausibility $\tilde{F}$.

$$[1]\ \tilde{F}([w_1 w_2 \bullet w_3]) = \frac{C([w_1 w_2 \bullet]) + C([w_2 \bullet w_3])}{C([w_1 w_2]) + C([w_2 w_3])},$$

where $\tilde{F}([w_1 w_2 \bullet w_3])$ is the plausibility that the position after a word sequence $w_1 w_2$ and before a word w_3 is a splitting position, $[w_l w_m]$ is a bigram, $[w_l w_m w_n]$ is a trigram, $\bullet$ indicates a boundary of sentences, and C(N-gram) means the appearance count of the N-gram in a training set.

It has also been reported that, for Japanese, three heuristics for Japanese part-of-speech and conjugation-type improve the performance. The heuristics indicate that the positions before and after particular part-of-speeches with particular conjugation types must or must not be splitting positions.

4 Applying Split-and-Translate to MT Systems

We apply the two split-and-translate methods to an MT system, D[3]. To apply method-N to an MT system is straightforward. When applying method-T, we consider the confidence factor of the MT system for criterion-3, rather as an optional criterion.

4.1 D[3] Overview

D[3] (Sumita, 2001) is an EBMT whose language resources are [i] a bilingual corpus, in which sentences are aligned beforehand; [ii] a bilingual dictionary, which is used for word alignment and generating target sentences; and [iii] thesauri of both languages, which are used for aiding word alignment and incorporating the semantic distance between words into the word sequence distance.

D[3] retrieves the most similar source sentence of examples from a bilingual corpus. For this purpose, DP-matching is used, which tells us the distance between

[3] Its English translation is "I see. And also how about breakfast?" in a corpus

word sequences, *dist*, while giving us the matched portions between the input and the example. *dist* is calculated as equation [2]. The counts of Insertion (I), Deletion (D), and substitution operations are summed. Then, this total is normalized by the sum of the lengths of the source and example sequences. Substitution is considered the semantic distance between two substituted words, or *SEMDIST*, which is defined using a thesaurus and ranges from 0 to 1.

$$[2]\ dist = \frac{I + D + 2\sum SEMDIST}{L_{input} + L_{example}}$$

The characteristics of D^3, especially in comparison with most EBMT proposals, are a) D^3 does not assume syntactic parsing and bilingual tree-banks; b) D^3 generates translation patterns on the fly according to the input and the retrieved translation examples as needed; c) D^3 uses examples sentence-by-sentence and does not combine examples.

Because of c), D^3's result is pretty good when a similar example is retrieved, but very bad otherwise. Therefore, we usually decide a threshold. If there is no example whose *dist* is within the given threshold, we must give up performing translation.

In an experiment using Basic Travel Expression Corpus (BTEC, described as BE-corpus in Takezawa, 2002), D^3's translation quality is very high. The experiment also shows a clear correlation between *dist* and the quality of translation. In other words, the accuracy decreases as the *dist* increases. In particular, the longer input sentences are, the more difficult for D^3 to find examples with a small *dist*.

4.2 Applying Method-T to D^3

As there is a correlation between *dist* and the translation quality, we can make use of *dist* as a confidence factor. To make the *combined-reliability*, each partial translation is weighted with its source word's number. That is, for each partial translation, its *dist* is multiplied by its source portion's length, and the resulting values are summed.

$$[3]\ combined\ reliability = \sum dist_{portion} \times L_{portion}$$

Adapting to D^3, criterion-3 is instantiated by the *combined-reliability* defined in equation [3].

5 Experiment

5.1 Preliminary

Target Systems

We investigated the two split-and-translate methods using D^3 in Japanese-to-English translation. We used a Japanese-and-English bilingual corpus, BTEC as the training set for D^3 and the Japanese part of BTEC as that for pre-process-splitting method for method-N. BTEC is a collection of Japanese sentences and their English translations usually found in phrase-books for foreign tourists. The statistics of the corpus is shown in Table 1.

Regarding D^3, the threshold for *dist* is 1/3.

For the pre-process-splitting method of method-N, the combinations of the parameters were used: 1) whether the heuristics for Japanese are used or not; 2) the threshold of splitting plausibility. The best results were selected from among the combinations in subsections 5.3 and 5.5.

Table 1. Corpus Statistics

	Japanese	English
# of sentences	152,172	
# of words	1,039,482	890,466
Vocabulary size	18,098	11,690
Average sentence length	6.83	5.85

Evaluation

The target is Japanese-to-English translation in this experiment. We extracted a test set from Bilingual Travel Conversation Corpus of Spoken Language (TC-corpus, Takezawa, 2002). All of the contents of TC-corpus are transcriptions of spoken dialogues between Japanese and English speakers through human interpreters. The test set of this experiment is 330 Japanese sentences from TC-corpus including no sentences spoken by the interpreters. The average length of the sentences in the test set is 11.4 (words). Therefore, the test sentences used in this experiment are much longer than the sentences in the training set, BTEC.

In this experiment, each translation result is graded into one of four ranks (described below) by a bilingual human translator who is a native speaker of the target language, American English:

(A) Perfect: no problem in either information or grammar;
(B) Fair: easy-to-understand with some unimportant information missing or flawed grammar;
(C) Acceptable: broken but understandable with effort;

(D) Nonsense: important information has been translated incorrectly (Sumita, 1999).

Adding to the four ranks, we use FAIL, or F, to indicate that there is no output sentence.

5.2 Translation without Splitting

Translations of the test set by D^3 without splitting were performed. The coverage of the output is lower. For 127 sentences, D^3 cannot yield results. The average length of the 127 sentences is 15.6. Afterward, we used these 127 sentences as the test set for split-and-translate methods.

5.3 Pre-Process-Splitting Quality

Before evaluating translation qualities of split-and-translate methods, we calculated the quality of the pre-process-splitting method of method-N on the 127 sentences. The positions where periods were manually inserted were regarded as the correct splitting positions. In the manual splitting process, they put periods at positions considered both grammatically and semantically adequate. There were 60 splitting positions, and 79 sentences, accounting for 62% of the 127 sentences, had no splitting position. Table 2 shows the numbers of sentences corresponding to those of splitting positions in a sentence.

Table 2. Number of splitting positions in a sentence vs. total number of sentences

# of split positions	0	1	2	3
# of sentences	79	37	10	1

The evaluation measure is based on how closely the result of the method corresponds to the correct solution, that is, recall and precision. We got a good result. The count of inferred positions is 65 in total, in which 55 positions are correct and 10 are incorrect, that is, recall is 91.7% and precision is 84.6%.

We also conducted an experiment on method-T as a method for only splitting sentences, extracting partial-translation boundaries. The result was bad: The count of inferred positions is 277 in total, in which 28 positions are correct and 249 are incorrect, that is, recall is 46.7% and precision is 10.1%. Although a smaller number of splittings is preferred with method-T, when most of the translations of long portions fail, method-T results in over-splitting.

The results show that the performance of method-N is much better than that of method-T when the target is only to split sentences.

5.4 Translation Quality of Method-T

Applying method-T to D^3, we performed translations of the 127 sentences by D^3. Table 3 shows the results, the number of each evaluation rank and the rate of the total number for each rank and better ranks than itself. As shown in the table, the rate of output is 100%, and the rate of success, which means that the rank is A, B or C, is 42.5%.

Table 3. Number and percentage of each Rank (Method-T)

A (A)	B (A+B)	C (A+B+C)	D (A+B+C+D)	F
4 (3.1%)	16 (15.7%)	34 (42.5%)	73 (100%)	0

There are correlations between quality ranks and *fault-length* or *partial-translation-count*. When the ratio of the *fault-length* to the entire input length is greater than 40% or the *partial-translation-count* is greater than 4, no result is successful.

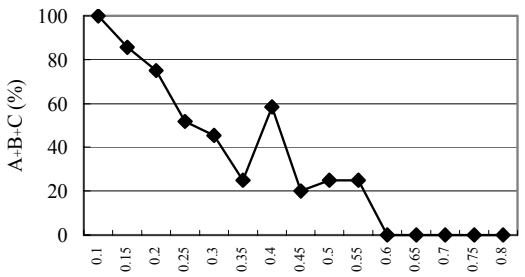

Figure 1. Success rate and *dist-in-splitting*

Furthermore, we can observe a correlation between success rate and *dist-in-splitting* in Figure 1. *dist-in-splitting* is defined by equation [4], an extension of *dist*, and ranges from 0 to 1. These correlations can give us a confidence factor on split-and-translate results.

$$[4]\ dist\ in\ splitting = \frac{\sum dist_{portion} \times L_{portion}}{L_{input}},$$

where $dist_{portion} = 1.0$ when the portion cannot be translated.

5.5 Translation Quality of Method-N

Applying method-N to D^3, we performed translations of the 127 sentences by D^3. Table 4 shows the results, which give the largest rate of success among the combinations of the parameters.

Table 4. Number and percentage of each Rank (Method-N)

A (A)	B (A+B)	C (A+B+C)	D (A+B+C+D)	F
4 (3.1%)	7 (8.7%)	16 (21.3%)	85 (88.2%)	15

The condition that is good for sentence splitting quality is not good for split-and-translate quality. On the condition of the parameters that gave the recall of 91.7% and the precision of 84.6%, the rate of output was 41.7% and that of success 6.3%. According to the correct splitting solution, among the 127 sentences that D^3 fails to translate without splitting, 79 sentences have no splitting position. Therefore, a good splitting for recall and precision has low probabilities for the rate of output and that of success. Put simply, when the threshold is smaller, although precision is worse, the rate of output and that of success are larger. However, the rates are much lower than those of method-T's results.

5.6 Summary of Experiments

Table 5. Splitting Quality and Split-and-Translate Quality

	Splitting		Split-and-Translate	
	recall	precision	success rate	output rate
Method-T	46.7%	10.1%	42.5%	100.0%
Method-N	91.7%	84.6%	21.3%	88.2%

Table 5 shows the summary of experiments. Though method-N is better in sentence splitting quality, method-T is better in split-and-translate quality.

6 Concluding Remarks

We have proposed a split-and-translate method and shown its effect through experiments. However, much more work remains to be accomplished.

To Improve Accuracy

The proposed method is based on three criteria. Although we have shown one combination of the criteria, there may be better combinations. Another possibility might be to integrate our method with another pre-process-splitting method, for example, by giving higher priorities to splitting positions as the latter method implies, which can be also used to improve the efficiency discussed below.

For Efficiency

Let N be the length of an input sentence, a naive implementation must search the solution in 2^{N-1} combinations, while trying $(N+1)N/2$ kinds of partial translations. However, there are several ways to optimize the algorithm. For example, it can be regarded as a shortest path problem, where each portion is an arc and portions without translations have high costs. There are effective algorisms for a shortest path problem. In addition, when the quality of translation has correlations with *fault-length*, *partial-translation-count*, and *dist-in-splitting*, as observed in subsection 5.4, candidates can be pruned by placing constraints on these factors.

Acknowledgements

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

Takezawa, T. et al. 2002. Toward a Broad-coverage Bilingual Corpus for Speech Translation of Travel Conversations in the Real World, Proc. of LREC-2002

Sumita, E. et al. 1999 Solutions to Problems Inherent in Spoken-language Translation: The ATR-MATRIX Approach, Proc. of MT Summit VII

Berger, A.L. et al. 1996. A Maximum Entropy Approach to Natural Language Processing, Association for Computatial Linguistics

Lavie, A. et al. 1996. Input Segmentation of Spontaneous Speech in JANUS: a Speech-to-speech Translation System, Proc. of ECAI-96 Workshop on Dialogue Processing in Spoken Language Systems

Takezawa, T. et al. 1999. Transformation into Meaningful Chunks by Dividing or Connecting Utterance Units, Journal of Natural Language Processing, Vol. 6 No. 2 (in Japanese)

Nakajima, H. et al. 2001. The Statistical Language Model for Utterance Splitting in Speech Recognition, Transactions of IPSJ, Vol. 42 No. 11 (in Japanese)

Kim, Y. B. et al. 1994. An Automatic Sentence Breaking and Subject Supplement Method for J/E Machine Translation, Transactions of IPSJ, Vol. 35 No. 6 (in Japanese)

Furuse, O. et al. 1998. Splitting Long or Ill-formed Input for Robust Spoken-language Translation, Proc. of COLING-ACL'98, pp. 421-427

Furuse, O. et al. 2001. Splitting Ill-formed Input for Robust Multi-lingual Speech Translation, Transactions of IPSJ, Vol. 42 No. 5 (in Japanese)

Wakita, Y. et al. 1997. Correct parts extraction from speech recognition results using semantic distance calculation, and its application to speech translation. Proc. of ACL/EACL Workshop on Spoken Language Translation, pp. 24-31

Sumita, E. 2001 Example-based machine translation using DP-matching between word sequences, Proc. of DDMT Workshop of 39th ACL

Sumita, E. 2002. Corpus-Centered Computation, ACL-02 Workshop on Speech-to-speech Translation, pp. 1-8