Splitting Long or Ill-formed Input for Robust Spoken-language Translation

Osamu FURUSE[†], **Setsuo YAMADA**, **Kazuhide YAMAMOTO**
ATR Interpreting Telecommunications Research Laboratories
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0288, Japan
furuse@cslab.kecl.ntt.co.jp, {syamada, yamamoto}@itl.atr.co.jp

Abstract

This paper proposes an input-splitting method for translating spoken-language which includes many long or ill-formed expressions. The proposed method splits input into well-balanced translation units based on a semantic distance calculation. The splitting is performed during left-to-right parsing, and does not degrade translation efficiency. The complete translation result is formed by concatenating the partial translation results of each split unit. The proposed method can be incorporated into frameworks like TDMT, which utilize left-to-right parsing and a score for a substructure. Experimental results show that the proposed method gives TDMT the following advantages: (1) elimination of null outputs, (2) splitting of utterances into sentences, and (3) robust translation of erroneous speech recognition results.

1 Introduction

A spoken-language translation system requires the ability to treat long or ill-formed input. An utterance as input of a spoken-language translation system, is not always one well-formed sentence. Also, when treating an utterance in speech translation, the speech recognition result which is the input of the translation component, might be corrupted even though the input utterance is well-formed. Such a misrecognized result can cause a parsing failure, and consequently, no translation output would be produced. Furthermore, we cannot expect that a speech recognition result includes punctuation marks such as a comma or a period between words, which are useful information for parsing. [1]

As a solution for treating long input, long-sentence splitting techniques, such as that of Kim (1994), have been proposed. These techniques, however, use many splitting rules written manually and do not treat ill-formed input. Wakita (1997) proposed a robust translation method which locally extracts only reliable parts, i.e., those within the semantic distance threshold and over some word length. This technique, however, does not split input into units globally, or sometimes does not output any translation result.

This paper proposes an input-splitting method for robust spoken-language translation. The proposed method splits input into well-balanced translation units based on a semantic distance calculation. The complete translation result is formed by concatenating the partial translation results of each split unit. The proposed method can be incorporated into frameworks that utilize left-to-right parsing and a score for a substructure, In fact, it has been added to Transfer-Driven Machine Translation (TDMT), which was proposed for efficient and robust spoken-language translation (Furuse, 1994; Furuse, 1996). The splitting is performed during TDMT's left-to-right chart parsing strategy, and does not degrade translation efficiency. The proposed method gives TDMT the following advantages: (1) elimination of null outputs, (2) splitting of utterances into sentences, and (3) robust translation of erroneous speech recognition results.

In the subsequent sections, we will first outline the translation strategy of TDMT. Then, we will explain the framework of our splitting method in Japanese-to-English (JE) and English-to-Japanese (EJ) translation. Next, by comparing the TDMT system's performance between two sets of translations with and without using the proposed method, we will demonstrate the usefulness of our method.

[†]Current affiliation is NTT Communication Science Laboratories.

[1]Punctuation marks are not used in translation input in this paper.

2 Translation strategy of TDMT

2.1 Transfer knowledge

TDMT produces a translation result by mimicking the example judged most semantically similar to the input string, based on the idea of Example-Based MT. Since it is difficult to store enough example sentences to translate every input, TDMT performs the translation by combining the examples of the partial expressions, which are represented by transfer knowledge patterns. Transfer knowledge in TDMT is compiled from translation examples. The following EJ transfer knowledge expression indicates that the English pattern "*X at Y*" corresponds to several possible Japanese expressions:

X at Y	=>	*Y' de X'*	((*present*, *conference*)..),
		Y' ni X'	((*stay*, *hotel*)..),
		Y' wo X'	((*look*, *it*)..)

The first possible translation pattern is "*Y' de X'*", with example set ((*present*, *conference*)..). We will see that this pattern is likely to be selected to the extent that the input variable bindings are semantically similar to the sample bindings, where *X* ="*present*" and *Y* ="*conference*". *X'* is the transfer result of *X*.

The source expression of the transfer knowledge is expressed by a constituent boundary pattern, which is defined as a sequence that consists of variables and symbols representing constituent boundaries (Furuse, 1994). A variable corresponds to some linguistic constituent. A constituent boundary is expressed by either a functional word or a part-of-speech bigram marker. In the case that there is no functional surface word that divides the expression into two constituents, a part-of-speech bigram is employed as a boundary marker, which is expressed by hyphenating the parts-of-speech of a left-constituent's last word and that of a right-constituent's first word.

For instance, the expression "*go to Kyoto*" is divided into two constituents, "*go*" and "*Kyoto*". The preposition "*to*" can be identified as a constituent boundary. Therefore, in parsing "*go to Kyoto*", we use the pattern "*X to Y*".

The expression "*I go*" can be divided into two constituents "*I*" and "*go*", which are a pronoun and a verb, respectively. Since there is no functional surface word between the two constituents, *pronoun-verb* can be inserted as a boundary marker into "*I go*", giving "*I pronoun-verb go*", which will now match the general transfer knowledge pattern "*X pronoun-verb Y*".

2.2 Left-to-right parsing

In TDMT, possible source language structures are derived by applying the constituent boundary patterns of transfer knowledge source parts to an input string in a left-to-right fashion (Furuse, 1996), based on a chart parsing method. An input string is parsed by combining active and passive arcs shifting the processed string left-to-right. In order to limit the combinations of patterns during pattern application, each pattern is assigned its linguistic level, and for each linguistic level, we specify the linguistic sublevels permitted to be used in the assigned variables.

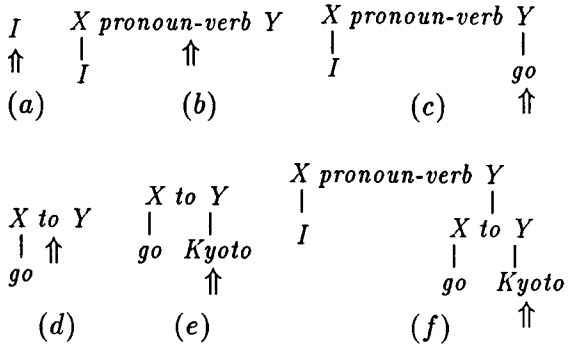

Figure 1: Substructures for "*I go to Kyoto*"

Figure 1 shows the substructures for each passive arc and each active arc in "*I go to Kyoto*". A processed string is indicated by "⇑". A passive arc is created from a content word shown in (a), or from a combination of patterns for which all of the variables are instantiated, like (c), (e), and (f). An active arc, which corresponds to an incomplete substructure, is created from a combination of patterns some of which have uninstantiated variables as right-hand neighbors to the processed string, like (b) and (d).

If the processed string creates a passive arc for a substring and the passive arc satisfies the leftmost part of an uninstantiated variable in the pattern of active arcs for the left-neighboring substring, the variable is instantiated with the passive arc. Suppose that the processed string is "*Kyoto*" in "*I go to Kyoto*". The passive arc (e) is created, and it instantiates *Y* of the active arc (b). Thus, by combining (b) and (e), the structure of "*I go to Kyoto*" is composed like (f). If a passive arc is generated in such operation, the creation of a new arc by variable instantiation is repeated. If a new arc can no longer be created, the processed string is shifted

to the right-neighboring string. If the whole input string can be covered with a passive arc, the parsing will succeed.

2.3 Disambiguation

The left-to-right parsing determines the best structure and best transferred result locally by performing structural disambiguation using semantic distance calculations, in parallel with the derivation of possible structures (Furuse, 1996). The best structure is determined when a relative passive arc is created. Only the best substructure is retained and combined with other arcs. The best structure is selected by computing the total sum of all the possible combinations of the partial semantic distance values. The structure with the smallest total distance is chosen as the best structure. The semantic distance is calculated according to the relationship of the positions of the words' semantic attributes in the thesaurus (Sumita, 1992).

3 Splitting strategy

If the parsing of long or ill-formed input is only undertaken by the application of stored patterns, it often fails and generates no results. Our strategy to parse such input, is to split the input into units each of which can be parsed and translated, and is explained as items (A)–(F) in this section.

3.1 Concatenation of neighboring substructures

The splitting is performed during left-to-right parsing as follows:

(A) Neighboring passive arcs can create a larger passive arc by concatenating them.

(B) A passive arc which concatenates neighboring passive arcs can be further concatenated with the right-neighboring passive arc.

These items enable two neighboring substructures to compose a structure even if there is no stored pattern which combines them. Figure 2 shows structure composition from neighboring substructures based on these items. α, β, and γ are structures of neighboring substrings. The triangles express substructures composed only from stored patterns. The boxes express substructures produced by concatenating neighboring substructures. δ is composed from its neighboring substructures, i.e., α and β. In addition, ϵ is composed from its neighboring substructures, i.e., δ and γ.

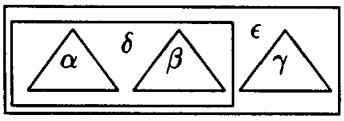

Figure 2: Structure from split substructures

Items (A) and (B) enable such a colloquial utterance as (1) to compose a structure by splitting, as shown in Figure 3.

(1) *"Certainly sir for how many people please"*

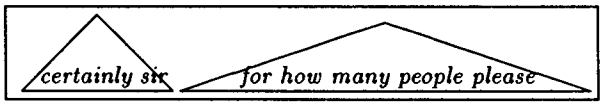

Figure 3: Structure for (1)

3.2 Splitting input into well-formed parts and ill-formed parts

Item (C) splits input into well-formed parts and ill-formed parts, and enables parsing in such cases where the input is ill-formed or the translation rules are insufficient. The well-formed parts can be applied patterns or they can consist of one content word. The ill-formed parts, which consist of one functional word or one part-of-speech bigram marker, are split from the well-formed parts.

(C) In addition to content words, boundary markers, namely, any functional words and inserted part-of-speech bigram markers, also create a passive arc and compose a substructure.

(2) *"They also have tennis courts too plus a disco"*

(3) *"Four please two children two adults"*

Suppose that the substrings of utterance (2), *"they also have tennis courts too"* and *"a disco"*, can create a passive arc, and that the system has not yet learned a pattern to which preposition *"plus"* is relevant, such as *"X plus Y"* or *"plus X"*.

Also, suppose that the substrings of utterance (3), *"four please"* and *"two children two adults"*, can create a passive arc, that part-of-speech

bigram marker "*adverb-numeral*" is inserted between these substrings, and that the system does not know pattern "*X adverb-numeral Y*" to combine a sentence for *X* and a noun phrase for *Y*.

By item (C), utterances (2) and (3) can be parsed in these situations as shown in Figure 4.

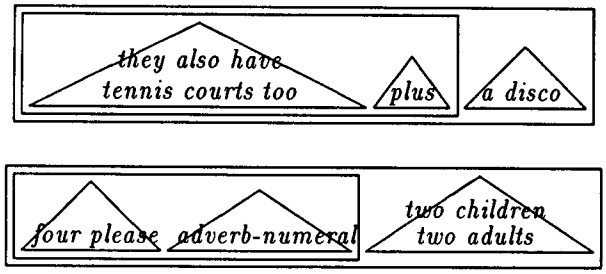

Figure 4: Structures for (2) and (3)

3.3 Structure preference

Although the splitting strategy improves robustness of the parsing, heavy dependence on the splitting strategy should be avoided. Since a global structure has more syntactic and semantic relations than a set of fragmental expressions, in general, the translation of a global expression tends to be better than the translation of a set of fragmental expressions. Accordingly, the splitting strategy should be used as a backup function.

Figure 5 shows three possible structures for "*go to Kyoto*". (a) is a structure relevant to pattern "*X to Y*" at the verb phrase level. In (b), the input string is split into two substrings, "*go*" and "*to Kyoto*". In (c), the input string is split into three substrings, "*go*", "*to*", and "*Kyoto*". The digit described at the vertex of a triangle is the sum of distance values for that strucure.

Among these three, (a), which does not use splitting, is the best structure. Item (D) is regulated to give low priority to structures including split substructures.

(D) When a structure is composed by splitting, a large distance value is assigned.

In the TDMT system, the distance value in each variable varies from 0 to 1. We experimentally assigned the distance value of 5.00 to one application of splitting, and 0.00 to the structure including only one word or one part-of-speech bigram marker. [2]

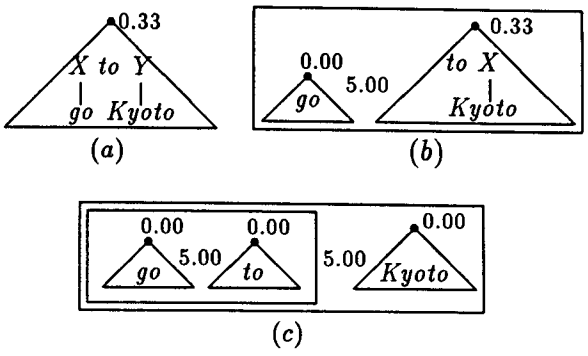

Figure 5: Structures for "*go to Kyoto*"

Suppose that substructures in Figure 5 are assigned the following distance values. The total distance value of (a) is 0.33. The splitting is applied to (b) and (c), once and twice, respectively. Therefore, the total distance value of (b) is 0.00+0.33+5.00×1=5.33, and that of (c) is 0.00+0.00+0.00+5.00×2=10.00. (a) is selected as the best structure because it gives the smallest total distance value.

3.4 Translation output

The results gained from a structure corresponding to a passive arc can be transferred and a partial translation result can then be generated. The translation result of a split structure is formed as follows:

(E) The complete translation result is formed by concatenating the partial translation results of each split unit.

A punctuation mark such as "," can be inserted between partial translation results to make the complete translation result clear, although we cannot expect punctuation in an input utterance. The EJ translation result of utterance (1) is as follows:

certainly sir | *for how many people please*
⇓ ⇓
hai , *nan-nin desuka*

Strings such as functional words and part-of-speech bigram markers have no target expression, and are transferred as follows:

[2] These values are tentatively assigned through comparing the splitting performance for some values, and are effective only for the present TDMT system.

Table 1: Effect of splitting on translation performance

	output rate (%)		parsing success rate (%)		output understandability (%)	
	w/o splitting	w/ splitting	w/o	w/	w/o	w/
JE	95.8	100	75.5	76.7	71.8	75.9
EJ	94.2	100	75.0	76.0	81.0	84.0
JK	83.4	100	68.3	71.2	80.4	94.5
KJ	66.7	100	54.1	56.4	64.1	90.5

(F) A string which does not have a target expression, is transferred to a string as "..", which means an incomprehensible part.

The EJ translation results of utterances (2) and (3) are as follows. "|" denotes a splitting position.

they also have tennis courts too | plus | a disco
⇓ ⇓ ⇓
douyouni tenisu-kooto ga mata ari-masu, .. , disuko

four please | adverb-numeral | two children two adults
⇓ ⇓ ⇓
yon o-negai-shi masu , .. , kodomo futari otona futari

4 Effect of splitting

The splitting strategy based on items (A)–(F) in Section 3 can be introduced to frameworks such as TDMT, which utilize left-to-right parsing and a score for a substructure. We discuss the effect of splitting by showing experimental results of the TDMT system's JE, EJ, Japanese-to-Korean (JK), and Korean-to-Japanese (KJ) translations. [3] The TDMT system, whose domain is travel conversations, presently can treat multi-lingual translation. The present vocabulary size is about 13,000 words in JE and JK, about 7,000 words in EJ, and about 4,000 words in KJ. The number of training sentences is about 2,900 in JE and EJ, about 1,400 in JK, and about 600 in KJ.

4.1 Null-output elimination

It is crucial for a machine translation system to output some result even though the input is ill-formed or the translation rules are insufficient. Items (C) and (D) in Section 3, split input into well-formed parts and ill-formed parts so that well-formed parts can cover the input as widely as possible. Since a content word and a pattern can be assigned some transferred results, some translation result can be produced if the input has at least one well-formed part.

Table 1 shows how the splitting improves the translation performance of TDMT. More than 1,000 sentences, i.e., new data for the system, were tested in each kind of translation. There was no null output, and a 100 % output rate in every translation. So, by using the splitting method, the TDMT can eliminate null output unless the morphological analysis gives no result or the input includes no content word. The splitting also improves the parsing success rate and the understandability of the output in every translation.

The output rates of the JK and KJ translations were small without splitting because the amount of sample sentences is less than that for the JE and EJ translations. However, the splitting compensated for the shortage of sample sentences and raised the output rate to 100 %. Since Japanese and Korean are linguistically close, the splitting method increases the understandable results for JK and KJ translations more than for JE and EJ translations.

4.2 Utterance splitting into sentences

In order to gain a good translation result for an utterance including more than one sentence, the utterance should be split into proper sentences. The distance calculation mechanism aims to split an utterance into sentences correctly.

(4) *"Yes that will be fine at five o'clock we will remove the bed"*

For instance, splitting is necessary to translate utterance (4), which includes more than one sentence. The candidates for (4)'s structure are shown in Figure 6. The total distance value of (a) is 0.00+1.11+5.00×1=6.11, that of (b) is 0.00+0.00+1.11+5.00×2=11.11, and that of (c) is 0.83+0.00+0.42+5.00×2=11.25. As (a) has the smallest total distance, it is chosen as the best structure, and this agrees with our intuition.

[3] In the experimental results referred to later in this section, the input does not consist of strings but of correct morpheme sequences. This enables us to focus on the evaluation of our splitting method by excluding cases where the morphological analysis fails.

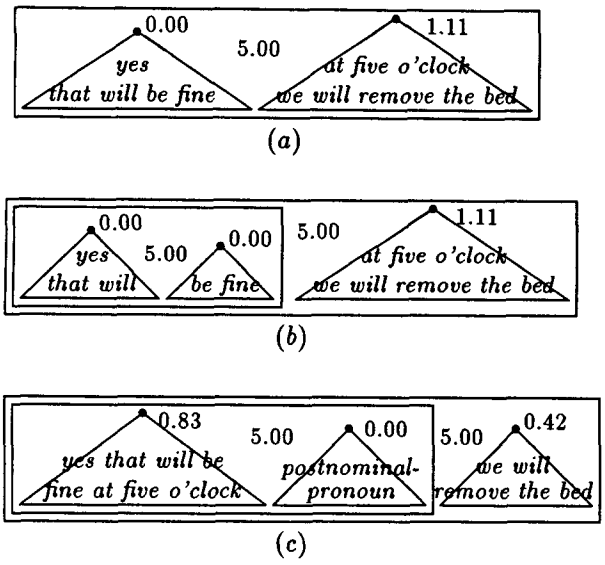

Figure 6: Structures for (4)

We have checked the accuracy of utterance splitting by using 277 Japanese utterances and 368 English utterances, all of which included more than one sentence. Table 2 shows the success rates for splitting the utterances into sentences. Although TDMT can also use the pattern "*X boundary Y*" in which *X* and *Y* are at the sentence level to split the utterances, the proposed splitting method increases the success rates for splitting the utterances in both languages.

Table 2: Success rates for splitting utterances

	w/o splitting	w/ splitting
Japanese	75.8	83.8
English	59.2	69.3

4.3 Translation after speech recognition

Speech recognition sometimes produces inaccurate results from an actual utterance, and erroneous parts often provide ill-formed translation inputs. However, our splitting method can also produce some translation results from such misrecognized inputs and improve the understandability of the resulting speech-translation.

Table 3 shows an example of a JE translation of a recognition result including a substitution error. The underlined words are misrecognized parts. "*youi*(preparation)" in the utterance is replaced with "*yori*(postposition)".

Table 4 shows an example of a JE translation of a recognition result including an insertion error. "*wo*" has been inserted into the utterance after speech recognition. The translation of the speech recognition result, is the same as that of the utterance except for the addition of ".."; ".." is the translation result for "*wo*", which is a postposition mainly signing an object.

Table 5 shows an example of the EJ translation of a recognition result including a deletion error. "*'s*" in the utterance is deleted after speech recognition. In the translation of this result, ".." appears instead of "*wa*", which is a postposition signing topic. ".." is the translation for marker "*pronoun-adverb*", which has been inserted between "*that*" and "*all*". The recognition result is split into three parts "*yes that*", "*pronoun-adverb*", and "*all correct*". Although the translations in Tables 3, 4, and 5 might be slightly degraded by the splitting, the meaning of each utterance can be communicated with these translations.

We have experimented the effect of splitting on JE speech translation using 47 erroneous recognition results of Japanese utterances. These utterances have been used as example utterances by the TDMT system. Therefore, for utterances correctly recognized, the translations of the recognition results should succeed. The erroneous recognition results were collected from an experimental base using the method of Shimizu (1996).

Table 6 shows the numbers of sentences at each level based on the extent that the meaning of an utterance can be understood from the translation result. Without the splitting, only 19.1% of the erroneous recognition results are wholly or partially understandable. The splitting method increases this rate to 57.4%. Failures in spite of the splitting are mainly caused by the misrecognition of key parts such as predicates.

Table 6: Translation after erroneous recognition

	w/o splitting	w/ splitting
wholly understandable	6 (12.8%)	15 (31.9%)
partially understandable	3 (6.3%)	12 (25.5%)
misunderstood, or never understandable	6 (12.8%)	20 (42.6%)
null output	32 (68.1%)	0 (0.0%)

4.4 Translation time

Since our splitting method is performed under left-to-right parsing, translation efficiency is not

Table 3: Substitution error in JE translation

	translation input	TDMT system's translation result
utterance	*Chousyoku no go youi wa deki masu ga*	*We can prepare breakfast.*
recognition result	*Chousyoku no go yori wa deki masu ga*	*Breakfast .. .. we can do.*

Table 4: Insertion error in JE translation

	translation input	TDMT system's translation result
utterance	*Soreto yoyaku ga hitsuyou desu ka*	*And is a reservation necessary?*
recognition result	*Soreto wo yoyaku ga hitsuyou desu ka*	*And .. is a reservation necessary?*

Table 5: Deletion error in EJ translation

	translation input	TDMT system's translation result
utterance	*Yes that 's all correct*	*Hai sore wa mattaku tadashii desu.*
recognition result	*Yes that all correct*	*Hai sore .. mattaku tadashii desu.*

a serious problem. We have compared EJ translation times in the TDMT system for two cases. One was without the splitting method, and the other was with it. Table 7 shows the translation time of English sentences with an average input length of 7.1 words, and English utterances consisting of more than one sentence with an average input length of 11.4 words. The translation times of the TDMT system written in LISP, were measured using a Sparc10 workstation.

Table 7: Translation time of EJ

input	w/o splitting	w/ splitting
sentence	0.35sec	0.36sec
utterance	0.60sec	0.61sec

The time difference between the two situations is small. This shows that the translation efficiency of TDMT is maintained even if the splitting method is introduced to TDMT.

5 Concluding remarks

We have proposed an input-splitting method for translating spoken-language which includes many long or ill-formed expressions. Experimental results have shown that the proposed method improves TDMT's performance without degrading the translation efficiency. The proposed method is applicable to not only TDMT but also other frameworks that utilize left-to-right parsing and a score for a substructure. One important future research goal is the achievement of a simultaneous interpretation mechanism for application to a practical spoken-language translation system. The left-to-right mechanism should be maintained for that purpose. Our splitting method meets this requirement, and can be applied to multi-lingual translation because of its universal framework.

References

O. Furuse and H. Iida. 1994. Constituent Boundary Parsing for Example-Based Machine Translation. In *Proc. of Coling '94*, pages 105–111.

O. Furuse and H. Iida. 1996. Incremental Translation Utilizing Constituent Boundary Patterns. In *Proc. of Coling '96*, pages 412–417.

Y.B. Kim and T. Ehara. 1994. An Automatic Sentence Breaking and Subject Supplement Method for J/E Machine Translation (in Japanese). In *Transactions of Information Processing Society of Japan*, Vol. 35, No. 6, pages 1018–1028.

T. Shimizu, H. Yamamoto, H. Masataki, S. Matsunaga, and Y. Sagisaka. 1996. Spontaneous Dialogue Speech Recognition using Cross-word Context Constrained Word Graphs. In *Proc. of ICASSP '96*, pages 145–148.

E. Sumita and H. Iida. 1992. Example-Based Transfer of Japanese Adnominal Particles into English. *IEICE Transactions on Information and Systems*, E75-D, No. 4, pages 585–594.

Y. Wakita, J. Kawai, and H. Iida. 1997. Correct parts extraction from speech recognition results using semantic distance calculation, and its application to speech translation. In *Proc. of ACL/EACL Workshop on Spoken Language Translation*, pages 24–31.