Splitting Input Sentence for Machine Translation Using Language Model with Sentence Similarity

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.jp

Abstract

In order to boost the translation quality of corpus-based MT systems for speech translation, the technique of splitting an input sentence appears promising. In previous research, many methods used N-gram clues to split sentences. In this paper, to supplement N-gram based splitting methods, we introduce another clue using sentence similarity based on edit-distance. In our splitting method, we generate candidates for sentence splitting based on N-grams, and select the best one by measuring sentence similarity. We conducted experiments using two EBMT systems, one of which uses a phrase and the other of which uses a sentence as a translation unit. The translation results on various conditions were evaluated by objective measures and a subjective measure. The experimental results show that the proposed method is valuable for both systems.

1 Introduction

We are exploring methods to boost the translation quality of corpus-based Machine Translation (MT) systems for speech translation. Among them, the technique of splitting an input sentence and translating the split sentences appears promising (Doi and Sumita, 2003).

An MT system sometimes fails to translate an input correctly. Such a failure occurs particularly when an input is long. In such a case, by splitting the input, translation may be successfully performed for each portion. 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 the splitting positions and the translations of the split portions are adequate, the possibility that the arrangement of the translations can provide an adequate translation of the complete input is relatively high. For example, the input sentence, "This is a medium size jacket I think it's a good size for you try it on please" [1] can be split into three portions, "This is a medium size jacket", "I think it's a good size for you" and "try it on please". In this case, translating the three portions and arranging the results in the same order give us the translation of the input sentence.

In previous research on splitting sentences, many methods have been based on word-sequence characteristics like N-gram (Lavie et al., 1996; Berger et al., 1996; Nakajima and Yamamoto, 2001; Gupta et al., 2002). 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 order to supplement sentence splitting based on word-sequence characteristics, this paper introduces another measure of sentence similarity. In our splitting method, we generate candidates for splitting positions based on N-grams, and select the best combination of positions by measuring sentence similarity. This selection is based on the assumption that a corpus-based MT system can correctly translate a sentence that is similar to a sentence in its training corpus.

The following sections describe the proposed splitting method, present experiments using two Example-Based Machine Translation (EBMT) systems, and evaluate the effect of introducing the similarity measure on translation quality.

2 Splitting Method

We define the term sentence-splitting as the result of splitting a sentence. A sentence-splitting is expressed as a list of sub-sentences that are

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

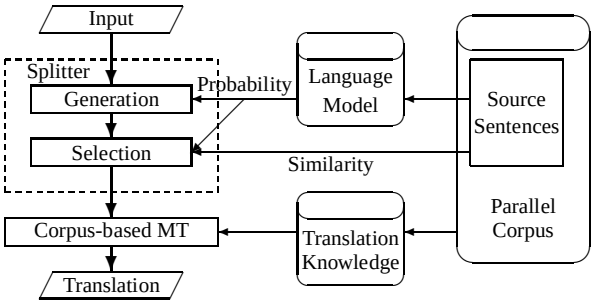

Figure 1: Configuration

portions of the original sentence. A sentence-splitting includes a portion or several portions. We use an N-gram Language Model (NLM) to generate sentence-splitting candidates, and we use the NLM and sentence similarity to select one of the candidates. The configuration of the method is shown in Figure 1.

2.1 Probability Based on N-gram Language Model

The probability of a sentence can be calculated by an NLM of a corpus. The probability of a sentence-splitting, $Prob$, is defined as the product of the probabilities of the sub-sentences in equation (1), where P is the probability of a sentence based on an NLM, S is a sentence-splitting, that is, a list of sub-sentences that are portions of a sentence, and P is applied to the sub-sentences.

$$Prob(S) = \prod_{s \in S} P(s) \quad (1)$$

To judge whether a sentence is split at a position, we compare the probabilities of the sentence-splittings before and after splitting. When calculating the probability of a sentence including a sub-sentence, we put pseudo words at the head and tail of the sentence to evaluate the probabilities of the head word and the tail word. For example, the probability of the sentence, "This is a medium size jacket" based on a trigram language model is calculated as follows. Here, p(z | x y) indicates the probability that z occurs after the sequence x y, and SOS and EOS indicate the pseudo words.

P(this is a medium size jacket) =
p(this | SOS SOS) ×
p(is | SOS this) ×
p(a | this is) ×
...
p(jacket | medium size) ×
p(EOS | size jacket) ×
p(EOS | jacket EOS)

This causes a tendency for the probability of the sentence-splitting after adding a splitting position to be lower than that of the sentence-splitting before adding the splitting position. Therefore, when we find a position that makes the probability higher, it is plausible that the position divides the sentence into sub-sentences.

2.2 Sentence Similarity

An NLM suggests where we should split a sentence, by using the local clue of several words among the splitting position. To supplement it with a wider view, we introduce another clue based on similarity to sentences, for which translation knowledge is automatically acquired from a parallel corpus. It is reasonably expected that MT systems can correctly translate a sentence that is similar to a sentence in the training corpus.

Here, the similarity between two sentences is defined using the edit-distance between word sequences. The edit-distance used here is extended to consider a semantic factor. The edit-distance is normalized between 0 and 1, and the similarity is 1 minus the edit-distance. The definition of the similarity is given in equation (2). In this equation, L is the word count of the corresponding sentence. I and D are the counts of insertions and deletions respectively. Substitutions are permitted only between content words of the same part of speech. Substitution is considered as the semantic distance between two substituted words, described as Sem, which is defined using a thesaurus and ranges from 0 to 1. Sem is the division of K (the level of the least common abstraction in the thesaurus of two words) by N (the height of the thesaurus) according to equation (3) (Sumita and Iida, 1991).

$$Sim_0(s_1, s_2) = 1 - \frac{I + D + 2\sum Sem}{L_{s_1} + L_{s_2}} \quad (2)$$

$$Sem = \frac{K}{N} \quad (3)$$

Using Sim_0, the similarity of a sentence-splitting to a corpus is defined as Sim in equation (4). In this equation, S is a sentence-splitting and C is a given corpus that is a set of sentences.

Sim is a mean similarity of sub-sentences against the corpus weighted with the length of each sub-sentence. The similarity of a sentence including a sub-sentence to a corpus is the greatest similarity between the sentence and a sentence in the corpus.

$$Sim(S) = \frac{\sum_{s \in S} L_s \cdot \max\{Sim_0(s,c) | c \in C\}}{\sum_{s \in S} L_s} \quad (4)$$

2.3 Generating Sentence-Splitting Candidates

To calculate Sim is similar to retrieving the most similar sentence from a corpus. The retrieval procedure can be efficiently implemented by the techniques of clustering (Cranias et al., 1997) or using A* search algorithm on word graphs (Doi et al., 2004). However, it still takes more cost to calculate Sim than $Prob$ when the corpus is large. Therefore, in the splitting method, we first generate sentence-splitting candidates by $Prob$ alone. In the generating process, for a given sentence, the sentence itself is a candidate. For each sentence-splitting of two portions whose $Prob$ does not decrease, the generating process is recursively executed with one of the two portions and then with the other. The results of recursive execution are combined into candidates for the given sentence. Through this process, sentence-splittings whose $Prob$s are lower than that of the original sentence, are filtered out.

2.4 Selecting the Best Sentence-Splitting

Next, among the candidates, we select the one with the highest score using not only $Prob$ but also Sim. We use the product of $Prob$ and Sim as the measure to select a sentence-splitting by. The measure is defined as $Score$ in equation (5), where λ, ranging from 0 to 1, gives the weight of Sim. In particular, the method uses only $Prob$ when λ is 0, and the method generates candidates by $Prob$ and selects a candidate by only Sim when λ is 1.

$$Score = Prob^{1-\lambda} \cdot Sim^{\lambda} \quad (5)$$

2.5 Example

Here, we show an example of generating sentence-splitting candidates with $Prob$ and selecting one by $Score$. For the input sentence, "This is a medium size jacket I think it's a good size for you try it on please", there may be many candidates. Below, five candidates, whose $Prob$ are not less than that of the original sentence, are generated. A '|' indicates a splitting position. The left numbers indicate the ranking based on $Prob$. The 5th candidate is the input sentence itself. For each candidate, Sim, and further, $Score$ are calculated. Among the candidates, the 2nd is selected because its $Score$ is the highest.

1. This is a medium size jacket | I think it's a good size for you try it on please
2. This is a medium size jacket | I think it's a good size for you | try it on please
3. This is a medium size jacket | I think it's a good size | for you try it on please
4. This is a medium size jacket | I think it's a good size | for you | try it on please
5. This is a medium size jacket I think it's a good size for you try it on please

3 Experimental Conditions

We evaluated the splitting method through experiments, whose conditions are as follows.

3.1 MT Systems

We investigated the splitting method using MT systems in English-to-Japanese translation, to determine what effect the method had on translation. We used two different EBMT systems as test beds. One of the systems was Hierarchical Phrase Alignment-based Translator (HPAT) (Imamura, 2002), whose unit of translation expression is a phrase. HPAT translates an input sentence by combining phrases. The HPAT system is equipped with another sentence splitting method based on parsing trees (Furuse et al., 1998). The other system was DP-match Driven transDucer (D^3) (Sumita, 2001), whose unit of expression is a sentence. For both systems, translation knowledge is automatically acquired from a parallel corpus.

3.2 Linguistic Resources

We used Japanese-and-English parallel corpora, i.e., a Basic Travel Expression Corpus (BTEC) and a bilingual travel conversation corpus of Spoken Language (SLDB) for training, and English sentences in Machine-Translation-Aided bilingual Dialogues (MAD) for a test set (Takezawa and Kikui, 2003). BTEC is a collection of Japanese sentences and their English translations usually found in phrase-books for foreign tourists. The contents of SLDB are transcriptions of spoken

dialogues between Japanese and English speakers through human interpreters. The Japanese and English parts of the corpora correspond to each other sentence-to-sentence. The dialogues of MAD took place between Japanese and English speakers through human typists and an experimental MT system.

(Kikui et al., 2003) shows that BTEC and SLDB are both required for handling MAD-type tasks. Therefore, in order to translate test sentences in MAD, we merged the parallel corpora, 152,170 sentence pairs of BTEC and 72,365 of SLDB, into a training corpus for HPAT and D^3. The English part of the training corpus was also used to make an NLM and to calculate similarities for the sentence-splitting method. The statistics of the training corpus are shown in Table 1. The perplexity in the table is word trigram perplexity.

The test set of this experiment was 505 English sentences uttered by human speakers in MAD, including no sentences generated by the MT system. The average length of the sentences in the test set was 9.52 words per sentence. The word trigram perplexity of the test set against the training corpus was 63.66.

We also used a thesaurus whose hierarchies are based on the Kadokawa Ruigo-shin-jiten (Ohno and Hamanishi, 1984) with 80,250 entries.

	English	Japanese
# of sentences	224,535	
# of words	1,589,983	1,865,298
avg. sentence length	7.08	8.31
vocabulary size	14,548	21,686
perplexity	27.58	27.37

Table 1: Statistics of the training corpus

3.3 Instantiation of the Method

For the splitting method, the NLM was the word trigram model using Good-Turing discounting. The number of split portions was limited to 4 per sentence. The weight of Sim, λ in equation (5) was assigned one of 5 values: 0, 1/2, 2/3, 3/4 or 1.

3.4 Evaluation

We compared translation quality under the conditions of with or without splitting. To evaluate translation quality, we used objective measures and a subjective measure as follows.

The objective measures used were the BLEU score (Papineni et al., 2001), the NIST score (Doddington, 2002) and Multi-reference Word Error Rate (mWER) (Ueffing et al., 2002). They were calculated with the test set. Both BLEU and NIST compare the system output translation with a set of reference translations of the same source text by finding sequences of words in the reference translations that match those in the system output translation. Therefore, achieving higher scores by these measures means that the translation results can be regarded as being more adequate translations. mWER indicates the error rate based on the edit-distance between the system output and the reference translations. Therefore, achieving a lower score by mWER means that the translation results can be regarded as more adequate translations. The number of references was 15 for the three measures.

In the subjective measure (SM), the translation results of the test set under different two conditions were evaluated by paired comparison. Sentence-by-sentence, a Japanese native speaker who had acquired a sufficient level of English, judged which result was better or that they were of the same quality. SM was calculated compared to a baseline. As in equation (6), the measure was the gain per sentence, where the gain was the number of won translations subtracted by the number of defeated translations as judged by the human evaluator.

$$\mathrm{SM} = \frac{\#\_\mathrm{of}\_\mathrm{wins} - \#\_\mathrm{of}\_\mathrm{defeats}}{\#\_\mathrm{of}\_\mathrm{test}\_\mathrm{sentences}} \qquad (6)$$

4 Effect of Splitting for MT

4.1 Translation Quality

Table 2 shows evaluations of the translation results of two MT systems, HPAT and D^3, under six conditions. In 'original', the input sentences of the systems were the test set itself without any splitting. In the other conditions, the test set sentences were split using $Prob$ into sentence-splitting candidates, and a sentence-splitting per input sentence was selected with $Score$. The weights of $Prob$ and Sim in the definition of $Score$ in equation (5) were varied from only $Prob$ to only Sim. The baseline of SM was the original.

The number of input sentences, which have multi-candidates generated with $Prob$, was 237, where the average and the maximum number of candidates were respectively 5.07 and 64. The average length of the 237 sentences was 12.79 words

		original	P^1S^0	$P^{1/2}S^{1/2}$	$P^{1/3}S^{2/3}$	$P^{1/4}S^{3/4}$	P^0S^1
# of split sentences		0	237	236	236	235	233
HPAT	BLEU	0.2979	0.3179	0.3201	0.3192	0.3193	0.3172
	NIST	7.1030	7.2616	7.2618	7.2709	7.2748	7.2736
	mWER	0.5828	0.5683	0.5665	0.5666	0.5658	0.5703
	SM		+6.9%	+8.7%	+10.1%	+10.1%	+9.5%
	# of wins		89	95	99	99	104
	# of defeats		54	51	48	48	56
	# of draws		94	90	89	88	73
D^3	BLEU	0.2992	0.3702	0.3704	0.3685	0.3695	0.3705
	NIST	2.1302	5.7809	5.8524	5.9115	5.9786	6.2545
	mWER	0.5844	0.5432	0.5433	0.5434	0.5424	0.5440
	SM		+20.6%	+21.8%	+21.8%	+22.4%	+23.0%
	# of wins		141	145	145	146	151
	# of defeats		37	35	35	33	35
	# of draws		59	56	56	56	47

Table 2: MT Quality: Using splitting vs. not using splitting, on the test set of 505 sentences (P indicates $Prob$ and S indicates Sim)

per sentence. The word trigram perplexity of the set of the 237 sentences against the training corpus was 73.87.

The table shows certain tendencies. The differences in the evaluation scores between the original and the cases with splitting are significant for both systems and especially for D^3. Although the differences among the cases with splitting are not so significant, SM steadily increases when using Sim compared to using only $Prob$, by 3.2% for HPAT and by 2.4% for D^3. Among objective measures, the NIST score corresponds well to SM.

4.2 Effect of Selection Using Similarity

Table 3 allows us to focus on the effect of Sim in the sentence-splitting selection. The table shows the evaluations on 237 sentences of the test set, where selection was required. In this table, the number of changes is the number of cases where a candidate other than the best candidate using $Prob$ was selected. The table also shows the average and maximum $Prob$ ranking of candidates which were not the best using $Prob$ but were selected as the best using $Score$. The condition of 'IDEAL' is to select such a candidate that makes the mWER of its translation the best value in any candidate. In IDEAL, the selections are different between MT systems. The two values of the number of changes are for HPAT and for D^3. The baseline of SM was the condition of using only $Prob$.

From the table, we can extract certain tendencies. The number of changes is very small when using both $Prob$ and Sim in the experiment. In these cases, the procedure selects the best candidates or the second candidates in the measure of $Prob$. Although the change is small when the weights of $Prob$ and Sim are equal, SM shows that most of the changed translations become better, some remain even and none become worse. The heavier the weight of Sim is, the higher the SM score is. The NIST score also increases especially for D^3 when the weight of Sim increases. The IDEAL condition overcomes most of the conditions as was expected, except that the SM score and the NIST score of D^3 are worse than those in the condition using only Sim. For D^3, the sentence-splitting selection with Sim is a match for the ideal selection.

So far, we have observed that SM and NIST tend to correspond to each other, although SM and BLEU or SM and mWER do not. The NIST score uses information weights when comparing the result of an MT system and reference translations. We can infer that the translation of a sentence-splitting, which was judged as being superior to another by the human evaluator, is more informative than the other.

4.3 Effect of Using Thesaurus

Furthermore, we conducted an experiment without using a thesaurus in calculating Sim. In the definition of Sim, all semantic distances of Sem

		P^1S^0	$P^{1/2}S^{1/2}$	$P^{1/3}S^{2/3}$	$P^{1/4}S^{3/4}$	P^0S^1	IDEAL
# of changes			10	19	25	91	111; 111
changed rank avg.			2.00	2.00	2.00	4.01	3.77; 3.78
(max)			(2)	(2)	(2)	(20)	(29); (23)
HPAT	BLEU	0.3004	0.3036	0.3022	0.3025	0.2994	0.3351
	NIST	7.1883	7.1911	7.2034	7.2068	7.1993	7.3057
	mWER	0.6363	0.6324	0.6328	0.6310	0.6405	0.5820
	SM		+3.4%	+3.8%	+3.8%	+5.9%	+14.8%
	# of wins		8	12	15	40	59
	# of defeats		0	3	6	26	24
	# of draws		2	4	4	25	28
D^3	BLEU	0.3310	0.3316	0.3291	0.3308	0.3340	0.3917
	NIST	6.0700	6.1687	6.2450	6.3372	6.6778	5.3250
	mWER	0.6181	0.6183	0.6185	0.6164	0.6197	0.5567
	SM		+3.4%	+3.4%	+5.5%	+6.3%	+5.5%
	# of wins		8	10	15	37	41
	# of defeats		0	2	2	22	28
	# of draws		2	7	8	32	42

Table 3: MT Quality: Using similarity vs. not using similarity, on the test set of 237 sentences (P indicates $Prob$ and S indicates Sim)

		P^1S^0	$P^{1/2}S^{1/2}$	$P^{1/3}S^{2/3}$	$P^{1/4}S^{3/4}$	P^0S^1	IDEAL
# of changes			10	19	26	93	111; 111
changed rank avg.			2.00	2.00	2.00	4.05	3.77; 3.78
(max)			(2)	(2)	(2)	(20)	(29); (23)
HPAT	BLEU	0.3004	0.3027	0.3034	0.3039	0.2973	0.3351
	NIST	7.1883	7.1830	7.1921	7.2003	7.1741	7.3057
	mWER	0.6363	0.6342	0.6320	0.6321	0.6346	0.5820
	SM		+1.7%	+3.8%	+3.4%	+6.3%	+14.8%
	# of wins		6	13	15	40	59
	# of defeats		2	4	7	25	24
	# of draws		2	2	4	28	28
D^3	BLEU	0.3310	0.3301	0.3310	0.3290	0.3370	0.3917
	NIST	6.0700	6.1387	6.2414	6.3341	6.6739	5.3250
	mWER	0.6181	0.6196	0.6188	0.6198	0.6175	0.5567
	SM		+3.0%	+4.6%	+5.9%	+7.6%	+5.5%
	# of wins		7	12	16	41	41
	# of defeats		0	1	2	23	28
	# of draws		3	6	8	29	42

Table 4: MT Quality: Using similarity vs. not using similarity, on the test set of 237 sentences, without a thesaurus (P indicates $Prob$ and S indicates Sim)

were assumed to be equal to 0.5. Table 4 shows evaluations on the 237 sentences.

Compared to Table 3, the SM score is worse when the weight of Sim in $Score$ is small, and better when the weight of Sim is great. However, the difference between the conditions of using or not using a thesaurus is not so significant.

5 Concluding Remarks

In order to boost the translation quality of corpus-based MT systems for speech translation, the technique of splitting an input sentence appears promising. In previous research, many methods used N-gram clues to split sentences. To supplement N-gram based splitting methods, we intro-

duce another clue using sentence similarity based on edit-distance. In our splitting method, we generate sentence-splitting candidates based on N-grams, and select the best one by the measure of sentence similarity. The experimental results show that the method is valuable for two kinds of EBMT systems, one of which uses a phrase and the other of which uses a sentence as a translation unit.

Although we used English-to-Japanese translation in the experiments, the method depends on no particular language. It can be applied to multilingual translation. Because the semantic distance used in the similarity definition did not show any significant effect, we need to find another factor to enhance the similarity measure. Furthermore, as future work, we'd like to make the splitting method cooperate with sentence simplification methods like (Siddharthan, 2002) in order to boost the translation quality much more.

Acknowledgements

The authors' heartfelt thanks go to Kadokawa-Shoten for providing the Ruigo-Shin-Jiten. 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

A.L. Berger, S. A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):1–36.

L. Cranias, H. Papageorgiou, and S. Piperidis. 1997. Example retrieval from a translation memory. *Natural Language Engineering*, 3(4):255–277.

G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. *Proc. of the HLT 2002 Conference*.

T. Doi and E. Sumita. 2003. Input sentence splitting and translating. *Proc. of Workshop on Building and Using Parallel Texts, HLT-NAACL 2003*, pages 104–110.

T. Doi, E. Sumita, and H. Yamamoto. 2004. Efficient retrieval method and performance evaluation of example-based machine translation using edit-distance (in Japanese). *Transactions of IPSJ*, 45(6).

O. Furuse, S. Yamada, and K. Yamamoto. 1998. Splitting long or ill-formed input for robust spoken-language translation. *Proc. of COLING-ACL'98*, pages 421–427.

N.K. Gupta, S. Bangalore, and M. Rahim. 2002. Extracting clauses for spoken language understanding in conversational systems. *Proc. of ICSLP 2002*, pages 361–364.

K. Imamura. 2002. Application of translation knowledge acquired by hierarchical phrase alignment for pattern-based mt. *Proc. of TMI-2002*, pages 74–84.

G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto. 2003. Creating corpora for speech-to-speech translation. *Proc. of EUROSPEECH*, pages 381–384.

A. Lavie, D. Gates, N. Coccaro, and L. Levin. 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*, pages 86–99.

H. Nakajima and H. Yamamoto. 2001. The statistical language model for utterance splitting in speech recognition (in Japanese). *Transactions of IPSJ*, 42(11):2681–2688.

S. Ohno and M. Hamanishi. 1984. *Ruigo-Shin-Jiten (in Japanese)*. Kadokawa, Tokyo, Japan.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. *RC22176, September 17, 2001, Computer Science*.

A. Siddharthan. 2002. An architecture for a text simplification system. *Proc. of LEC 2002*.

E. Sumita and H. Iida. 1991. Experiments and prospects of example-based machine translation. *Proc. of 29th Annual Meeting of ACL*, pages 185–192.

E. Sumita. 2001. Example-based machine translation using dp-matching between word sequences. *Proc. of 39th ACL Workshop on DDMT*, pages 1–8.

T. Takezawa and G. Kikui. 2003. Collecting machine-translation-aided bilingual dialogues for corpus-based speech translation. *Proc. of EUROSPEECH*, pages 2757–2760.

N. Ueffing, F.J. Och, and H. Ney. 2002. Generation of word graphs in statistical machine translation. *Proc. of Conf. on Empirical Methods for Natural Language Processing*, pages 156–163.