Simple, Lexicalized Choice of Translation Timing for Simultaneous Speech Translation

Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, Satoshi Nakamura

Graduate School of Information Science, Nara Institute of Science and Technology
{tomoki-f,neubig,ssakti,tomoki,s-nakamura}@is.naist.jp

Abstract

Conventional speech translation systems wait until the end of the input sentence before starting translation, causing a large delay in the translation process. Methods have been proposed to reduce this delay by dividing the input utterance on pause boundaries, but while these methods have proven useful on speech translation of language pairs with similar word order, they are insensitive to linguistic information and less effective for languages that require more word reordering. In this work, we propose a method that uses lexicalized information to perform translation unit segmentation considering the relationship between the source and target languages. In particular, we use the phrase table and reordering probabilities used in phrase-based translation systems to decide points in the sentence where we can begin translation with less delay. Through an experimental evaluation, we confirmed that the proposed method significantly reduces delay for Japanese-English and French-English translation. We also show that a parameter introduced in our model can adjust the trade-off between simultaneity and accuracy, and that in situations that require a large degree of simultaneity, our system can achieve a delay reduction of 20% compared to pause segmentation with identical accuracy.

Index Terms: speech translation, real-time, delay, simultaneous interpretation, sentence segmentation

1. Introduction

While translation accuracy of speech translation systems has been improved by years of research, it is still not possible to output translation results in real time. The reason for this lies in the interaction between the three components of conventional speech translation systems: automatic speech recognition (ASR), machine translation (MT) and text-to-speech (TTS). Normally, the MT module is started after the ASR module finishes recognition and the TTS module is started after MT module finishes translation, causing a delay between the start of the speaker's utterance to the end of synthesis.

Conventional speech translation systems use full sentences as the fundamental unit of processing [1]. As a result, the MT module is not able to start translation until the user finishes uttering the sentence, and longer sentences require more time for MT decoding. In contrast, human simultaneous interpreters generally break sentences into smaller chunks, resulting in a lower delay (or "ear-to-voice span") [2].

As a solution to this problem, we propose a method for starting the translation process before the sentence finishes, allowing the MT module to start translation more quickly and shorten processing time. This results in provision of information to the listener in closer-to-real time. As a concrete method to decide when to start the translation process before the user has entirely finished uttering a sentence, we use the phrase table used in phrase-based MT. The first element of the method consists of using phrase patterns of the source language to shorten the translation unit. Second, we introduce a parameter into the model that allows us to adjust the length of the translation unit based on the linguistic qualities of the translation pair at hand. This parameter specifies a threshold for each phrase's right probability (RP), which shows the degree to which the order of the source and target language can be expected to be the same. This allows us to improve translation accuracy by translating in longer units than phrases in the cases where phrases are too short to translate accurately.

In an experimental evaluation, we examine the effect that the proposed method has on translation accuracy and delay using translation between English and Japanese or French. The results of the evaluation confirm that the proposed method is able to reduce the delay in translation, and that the RP threshold is able to adjust the trade-off between simultaneity and accuracy. In situations that require a large degree of simultaneity, we also find that our system can achieve a delay reduction of 20% compared to pause segmentation with identical accuracy.

2. Related Work

There is not a large amount of previous research related to improving the simultaneity of speech translation, but there are a few previous works of note. The first is a method of determining the translation unit utilizing incremental dependency parsing and manually created rules based on these parses [3]. This method has the advantage of being able to incorporate human linguistic intuition, but is difficult to adapt to new language pairs because a human linguist must remake syntactic rules and requires an accurate syntactic parser for every language handled. Another recently proposed method determines the translation unit by using the silence interval of ASR [4]. This method has the advantage of being trivial to apply to any spoken language, as it only uses prosodic features. However, it also cannot determine the translation unit using linguistic information.

In the proposed method, we try to strike a balance between these two previous approaches by utilizing linguistic information that can be obtained from only a parallel corpus. This allows us to determine the translation unit in a way that is tenable to achieving higher translation accuracy while still allowing for easy adaption to new languages.

3. Proposed Method

We design our proposed method around the widely used framework of statistical phrase-based MT [5]. Within this framework, the phrase table specifies which source phrases can be translated into which target phrases, with an example shown in Table 1.

Table 1: *Phrase table and right probability (RP)*

Source	Target	RP
watashi	I	0.8
watashi ha	I	0.9
otoko	man	0.2
otoko desu	am a man	0.6
nan	what	0.9
nan ji	what time	0.7
na ji kara	from what time	0.5
pure-	play	0.2
deki	can	0.7
deki masuka	?	0.95

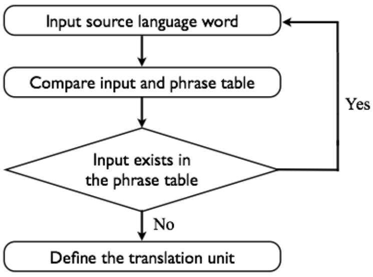
Input source language word
Compare input and phrase table
Input exists in the phrase table
Yes
No
Define the translation unit

Figure 1: *Deciding translation timing using the phrase table*

This table can be automatically extracted by conducting word alignment and phrase extraction over a parallel corpus of source and target sentences. We describe the RP shown in the third column in detail in Section 3.2. For the rest of this paper, we will assume that we have an already-trained phrase-based SMT system, and are using its phrase table to decide the appropriate timing with which to generate translations.

3.1. Deciding Translation Timing using the Phrase Table

Because the phrase table can be learned from only a parallel corpus, it can be created for any language for which we have parallel data. Thus, we chose to focus on using the phrase table as a simple and multilingual method to decide the appropriate timing for speech translation. Figure 1 shows the process of finding a translation unit using the phrase table.

Here, we assume $F = f_1 \dots f_J$ is the source language sentence and $G = g_1 \dots g_K$ is a cache of incoming words. We assume that F is input one word at a time, as is the case for a translation system consuming the input of a one-pass ASR decoder.[1] We decide at each word f_j whether to begin translation for f_j and all previous untranslated words, or wait until a later point in time to begin translation. For each word f_j, the process starts by adding the word to the end of G. Next, G is compared with source language patterns in the phrase table. If G matches at least one source language pattern in the phrase table, G is retained to be translated at a later time. If G does not match the phrase table, we send all but the final word in the cache ($g_1 \dots g_{K-1}$) to the translation engine and replace the cache so it only contains the final word $G \leftarrow g_K$.

The motivation for this method is that it allows us to perform monotonic phrase-based translation in real time. The cache will be expanded only as long as it hits a phrase in the phrase table, so we will translate in units that correspond to the longest phrase starting the as-of-yet-untranslated word string.

Table 2: *Segmentation result*

Unit	Result
watashi ha	I
otoko desu	am a man

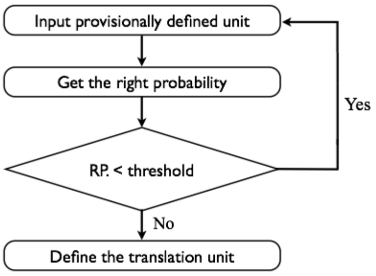
Input provisionally defined unit
Get the right probability
RP. < threshold
Yes
No
Define the translation unit

Figure 2: *Deciding translation timing using right probability*

Next we show a concrete example for Japanese-English translation given an input $F =$ *"watashi ha otoko desu"* ("I am a man"). We will use the example in Table 1 as our phrase table. First, f_1 is added to G and G becomes *"watashi"*. As *"watashi"* exists in the phrase table, we do not immediately translate. Next, we add f_2 to G, giving us $G =$ *"watashi ha"*, which also exists in the phrase table, so we leave the cache as-is. Next, we add f_3 to G, giving us $G =$ *"watashi ha otoko"*, which does not exist in the phrase table. So, we send $g_1 \dots g_{K-1}$ (*"watashi ha"*) to the translation engine, and replace the cache with g_K (*"otoko"*). The final result of the translation according to this scheme is shown in Table 2.

3.2. Adjusting Translation Timing using Right Probability

While the previously proposed method is able to perform monotonic translation, in most situations this is not sufficient to generate good translations. Phrases are able to capture some local reordering, but the monotonic constraint prevents us from reordering over phrase boundaries.

An example of a situation where this causes problems is shown in Table 3. In this example, we would like to translate *"pure- deki masu ka"* into "can we play". However, this whole sentence does not exist in the phrase table, and when we translate *"pure-"* (play) and *"deki masu ka"* (can) separately in monotonic order, we cannot achieve the proper translation because the word orders of the source language and target languages are different. Thus, if we are able to instead translate using a unit that allows the order of these phrases to change, translation accuracy can be expected to increase.

As a simple but effective method of judging for which phrases a reordering is likely to occur, we propose using the *right probability* (RP) of each phrase. In reordering models for phrase-based MT systems [6], the RP is formally defined as the probability that when the current phrase ends at words f_j and e_i in the source and target respectively, the source phrase beginning at f_{j+1} is aligned to a target phrase starting at e_{i+1} or later.[2] In other words, The RP indicates the probability that the

[1] The situation where we accept not single words but short sequences of words at a time is a realistic and interesting target for future work.

[2] Reordering models generally use *monotone*, *swap*, and *discontinuous* probabilities. The RP covers all *monotone* reorderings, and *discontinuous* reorderings where the source and target are in the same order.

Table 3: *A failed translation using only the phrase table*

Unit	Result
nan ji kara	from what time
pure-	play
deki masu ka	?

Table 4: *BLEU of each LM for units defined with RP 0.0*

Translation Unit RP	LM RP	BLEU
0.0	0.0	38.46
0.0	1.0	34.04

phrases occur in the same order in both languages, and thus directly correlates to our goal of judging when we can segment the input sentence without disturbing translation order.

Figure 2 shows the process of choosing the translation unit using the RP. First, we define the unit provisionally using the method of in Section 3.1. Next, we compare the RP of the current longest phrase with a threshold, and if the threshold is exceeded we send the contents of the cache to the translator, and if the threshold is not exceeded, we save the words for translation at a later time. For example, if we set the threshold to 0.5, the RP of *"pure-"* is lower than 0.5, so, we do not output it immediately, instead waiting until we have *"pure- deki masu ka"* as the translation unit. In this framework, setting the threshold 1.0 is equivalent to conventional speech translation systems, setting the threshold to 0.0 is equivalent to method of Section 3.1.

3.3. Language Model Adaptation

In the two proposed methods, we focus on choosing the unit with which we translate. However, we also need to be careful about how we treat the language model (LM) probabilities. If we use a LM trained on full sentences, the translation result often includes useless punctuation at the end of each translated segment. This is due to the fact that most full sentences will end with punctuation, so LMs trained on full sentences will give very high values for conditional probabilities such as $P(w_i =$ "</s>"$|w_{i-1} =$ ".") and $P(w_i =$ "</s>"$|w_{i-1} =$ "?").

As a solution, we propose a method of creating a LM adapted to the translation unit. The idea is simple and only consists of splitting the target language sentences using RP according to the method of Section 3.2 before training the LM. In Table 4, we show results of a preliminary experiment justifying this method. We define the translation unit according to proposed method in Section 3.1, and the LM is trained with text split with a probability threshold of 0.0 (a phrase-level LM) or 1.0 (a sentence level LM). In this result, we can see that for the translation unit defined using a RP threshold of 0.0, the LM trained with the matched threshold is clearly better than the sentence-based LM.

It should be noted that [4] recently proposed a method for handling the LM that carries over the history from the previous translation result. This method has the advantage of not requiring re-training of the LM, but also has the disadvantage of requiring the translation result for the previous utterance before being able to start translation of the next utterance, making it impossible to translate multiple utterances in parallel. We leave a direct comparison of these two methods for future work.

Table 5: *Number of sentences and words in the experiment data*

ja-en	Sent.	Words(ja)	Words(en)
Training	162k	1.38M	1.19M
Test	1,018	8,782	7,496
Test (11+)	217	3,092	2,234
fr-en	**Sent.**	**Words(fr)**	**Words(en)**
Training	44k	1.02M	880k
Test	960	26,753	22,717

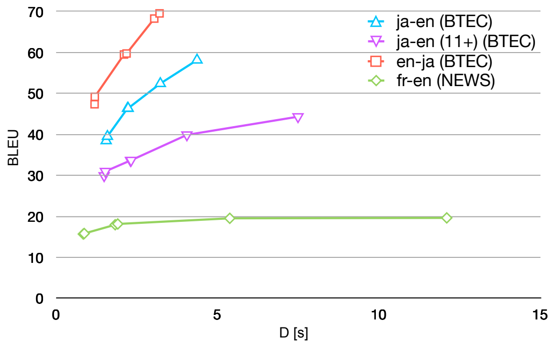

Figure 3: *Translation accuracy and delay on manual transcripts*

4. Experiment

4.1. Experiment Setup

While the final goal of our research is to improve speech translation, to focus on the effect of translation unit selection on translation speed and accuracy, we perform a simulation using transcripts created manually or using ASR. For the RP threshold, we use values of 0.0, 0.2, 0.4, 0.6, 0.8 and 1.0, with 1.0 being equivalent to a baseline of sentence-based translation. We use the same RP threshold for both translation unit selection and LM adaptation as mentioned in Section 3.3. We perform the majority of our experiments on Japanese-English (ja-en) translation. Because this task is difficult due to the large difference of word order, we also experiment with the more similar French-English (fr-en) pair. We also try translation in the English-Japanese (en-ja) direction to confirm whether there is a difference based on the direction of the translation. We use Julius [7] to perform speech recognition and Moses [8] to perform translation, and Mecab [9] to perform Japanese morphological analysis.

Table 5 shows the experimental data used from the Basic Travel Expression Corpus (BTEC) [10] for ja-en and en-ja, and NEWS [11] for fr-en. As the BTEC sentences are relatively short compared to NEWS, we also experiment with longer sentences that contain at least 11 words from BTEC.

For evaluation measures, we use BLEU [12] and measure translation accuracy with 12 reference for ja-en, and 1 reference for fr-en. We also perform a manual evaluation using a 0-5 scale based on acceptability [13]. We calculate translation delay D as $D = A + T$. A is the ASR time per sentence, and we calculate this using the time of each wave file in the test set. T indicates the average MT decoding time per sentence.

4.2. Translation Delay and Accuracy

We first show results of translation experiments on manual transcripts in Figure 3. According to this result, we can see that for

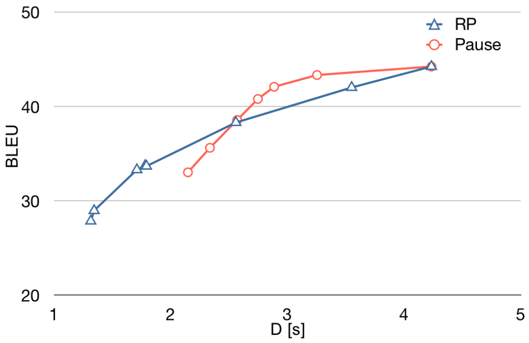

Figure 4: *Translation accuracy and delay for ASR using pause-based and RP-based unit selection (ja-en)*

all four settings as we reduce the RP threshold, translation delay decreases, at the cost of a drop in accuracy across all data sets.

First, we compare results of BTEC ja-en using the normal and long sentence test sets to investigate the effectiveness of the proposed method for longer sentences. From these results, we can see that the speed-accuracy curves are similar for ja-en and ja-en (11+). However, the amount that D decreases on the normal and the long test sets is notably different. In the normal test set, the delays are 4.36s and 1.55s for right probabilities 1.0 and 0.0, respectively, while for the long test set, delays are 7.49s and 1.48s. Given this result, it is likely that the proposed method is more effective for long sentences than shorter sentences.

Next, we investigate the tendency in case of the reversal of source and target language. Comparing ja-en and en-ja translation we confirmed the fact that both achieve similar speed-accuracy curves. In addition, BLEU is higher overall for en-ja because Japanese sentences are longer than English sentences, so the number of matches with the reference is greater than when the target language is English.

Finally, we compare ja-en and fr-en translation to investigate the effectiveness for a language pair with small difference of word order. As can be seen from the graph for fr-en, by reducing the RP threshold from 1.0 to 0.8 we are able to achieve a decrease in delay from 12.1s to 5.40s with a almost no drop in BLEU (19.63 to 19.53 respectively). Even when we set the threshold lower, the drop in accuracy is much smaller than ja-en or en-ja translation, confirming that the proposed method is particularly effective for languages with similar word order.

4.3. Experiments on ASR Results

Next, we show results using actual ASR output for ja-en. Figure 4 shows the results, both with the proposed method, and with pause based segmentation similar to [4]. Here, we use RP segmentation at 0.0, 0.2, 0.4, 0.6, 0.7, 0.8, 0.9, or 1.0, and use short-pause segmentation with 1, 2, 3, 4, 5, or 10 frames. For the LM in pause-based segmentation, we perform LM adaptation described in Section 3.3, and use the RP threshold 0.8, which provided the best results.

First comparing ASR with the use of manual transcripts, we can see that while BLEU is lower because of ASR errors, the speed/accuracy trade-off is similar to that of manual transcripts. Next, we experiment using the ASR pause-based method to compare to proposed method. From the results, we can see that the proposed method achieves higher accuracies when translating with very low delay, and is able to segment even when no prosodic pause exists while pause-based segmentation is effective for slower translation speeds. Compared with short-pause segmentation with a frame size of 1, which achieved a BLEU of 33.0 and a delay of 2.14s, RP segmentation with a threshold of 0.4 achieved a BLEU of 33.3 and a delay of 1.71s, a 20.0% decrease in delay with nearly identical accuracy.

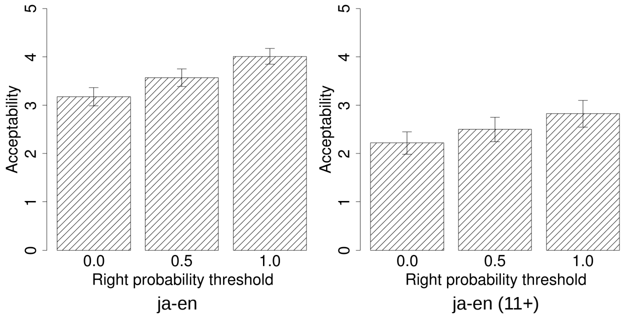

Figure 5: *Subjective evaluation of acceptability*

Table 6: *An example of a segmented sentence*

RP	Result	Acceptability
0.0	for surfing / tell me a good place /	5
1.0	please tell me a good surfing place ? /	5

4.4. Subjective Evaluation

In this section, we perform a manual evaluation of the translation results for RP thresholds of 0.0, 0.5 and 1.0. For the ja-en, a total of 300 sentences were scored by five evaluators. For ja-en (11+), a total of 160 sentences were scored.

Figure 5 shows subjective evaluation result of BTEC ja-en and ja-en (11+). From the results, we can see that while subjective evaluation decreases, the decrease is smaller than that of BLEU. This is because in many cases, even if the order is slightly unnatural, we still can achieve an understandable translation. An example of such a translation is shown in Table 6. This indicates that the development of automatic evaluation metrics specifically targeted for simultaneous translation is an interesting challenge for future research.

5. Conclusions

In this research, we proposed three methods for improving the simultaneity of speech translation that are both simple and applicable to any language for which we have parallel data. We found that the proposed method can decrease the start time and processing time of MT compared to a conventional ST. This speech translation system can be used in various situations, using a higher RP when more delay is acceptable, and a lower RP when highly simultaneous results are required.

There are a number of avenues for future work. In this work we translated ASR transcripts, but we must consider tighter integration with the ASR and TTS modules. It is also likely that we can further improve segmentation efficiacy by incorporating more sophisticated syntactic or prosodic information in our automatically learned rules. Finally, we hope to investigate automatic evaluation measures that show high correlation with human judgements in simultaneous interpretation situations, including a comparison with actual human interpreters.

Acknowledgment: Part of this work was supported by JSPS KAKENHI Grant Number 24240032.

6. References

[1] E. Matusov, A. Mauser, and H. Ney, “Automatic sentence segmentation and punctuation prediction for spoken language translation,” in *Proceedings of IWSLT*, 2006, pp. 158–165.

[2] F. Goldman-Eisler, “Segmentation of input in simultaneous translation,” *Journal of Psycholinguistic Research*, vol. 1, no. 2, pp. 127–140, 1972.

[3] K. Ryu, A. Mizuno, S. Matsubara, and Y. Inagaki, “Incremental Japanese spoken language generation in simultaneous machine interpretation,” in *Proceedings of Asian Symposium on Natural Language Processing to Overcome Language Barriers in Hainan Island China*, 2004.

[4] S. Bangalore, V. K. R. Sridhar, P. K. L. Golipour, and A. Jimenez, “Real-time incremental speech-to-speech translation of dialogs,” in *Proceedings of NAACL*, 2012.

[5] P. Koehn, F. Och, and D. Marcu, “Statistical phrase-based translation,” in *Proceedings of NAACL-HLT*, 2003, pp. 48–54.

[6] P. Koehn, A. Axelrod, A. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, “Edinburgh system description for the 2005 IWSLT speech translation evaluation,” in *Proceedings of IWSLT*, 2005.

[7] A. Lee, T. Kawahara, and K. Shikano, “Julius—an open source real-time large vocabulary recognition engine,” 2001.

[8] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens *et al.*, “Moses: Open source toolkit for statistical machine translation,” in *Proceedings of ACL*, 2007, pp. 177–180.

[9] T. Kudo, K. Yamamoto, and Y. Matsumoto, “Applying conditional random fields to japanese morphological analysis,” in *Proceedings of EMNLP*, 2004, pp. 230–237.

[10] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, “Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world,” in *Proceedings of LREC*, 2002, pp. 147–152.

[11] J. Civera and A. Juan, “Domain adaptation in statistical machine translation with mixture modelling,” in *Proceedings of WMT*, 2007, pp. 177–180.

[12] K. Papineni, S. Roukos, T. Ward, and W. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002, pp. 311–318.

[13] I. Goto, B. Lu, K. Chow, E. Sumita, and B. Tsou, “Overview of the patent machine translation task at the NTCIR-9 workshop,” in *Proceedings of NTCIR*, 2011, pp. 559–578.