Constructing a Speech Translation System using Simultaneous Interpretation Data

Hiroaki Shimizu, Graham Neubig, Sakriani Sakti,
Tomoki Toda, Satoshi Nakamura

Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma-shi, Nara, Japan
{hiroaki-sh,neubig,ssakti,tomoki,s-nakamura}@is.naist.jp

Abstract

There has been a fair amount of work on automatic speech translation systems that translate in real-time, serving as a computerized version of a simultaneous interpreter. It has been noticed in the field of translation studies that simultaneous interpreters perform a number of tricks to make the content easier to understand in real-time, including dividing their translations into small chunks, or summarizing less important content. However, the majority of previous work has not specifically considered this fact, simply using translation data (made by translators) for learning of the machine translation system. In this paper, we examine the possibilities of additionally incorporating simultaneous interpretation data (made by simultaneous interpreters) in the learning process. First we collect simultaneous interpretation data from professional simultaneous interpreters of three levels, and perform an analysis of the data. Next, we incorporate the simultaneous interpretation data in the learning of the machine translation system. As a result, the translation style of the system becomes more similar to that of a highly experienced simultaneous interpreter. We also find that according to automatic evaluation metrics, our system achieves performance similar to that of a simultaneous interpreter that has 1 year of experience.

1. Introduction

While the translation performance of automatic speech translation (ST) has been improving, there are still a number of areas where ST systems lag behind human interpreters. One is accuracy of course, but another is with regards to the speed of translation. When simultaneous interpreters interpret lectures in real time, they perform a variety of tricks to shorten the delay until starting the interpretation. There are two main techniques. The first technique, also called the *salami technique*, is to divide longer sentences up into a number of shorter ones, resulting in a lower delay [1]. The second technique is to adjust the word order of the target language sentence to more closely match the source language, especially for language pairs that have very different grammatical structure. An example of this that we observed in our data of English-Japanese translation and simultaneous interpretation is shown in Figure 1. When looking at the source and the translation, the word order is quite different, reversing two long clauses: A and B. In contrast, when looking at the source and the simultaneous interpretation, the word order is similar. If a simultaneous ST system attempts to reproduce the first word order, it will only be able to start translation after it has received the full "A because B." On the other hand, if the system is able to choose the word order closer to human interpreters, it can begin translation after "A," resulting in a lower delay.

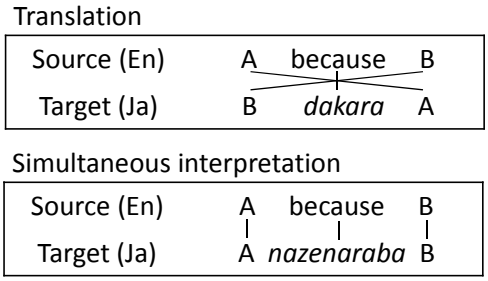

Figure 1: Difference between translation and simultaneous interpretation word order

There are several related works about simultaneous ST [2][3][4] that automatically divide longer sentences up into a number of shorter ones similarly to the salami technique employed by simultaneous interpreters. While these related works aim to segment sentences in a similar fashion to simultaneous interpreters, all previous works concerned with sentence segmentation have used translation data (made by translators) for learning of the machine translation system. In addition, while there are other related works about collecting simultaneous interpretation data [5][6][7], all previous works did not compare simultaneous interpreters of multiple experience levels and did investigate whether this data can be used to improve the simultaneity of actual MT systems.

In this work, we examine the potential of simultaneous interpretation data (made by simultaneous interpreters) to

Table 1: Profile of simultaneous interpreters

Experience	Rank	Lectures	Minutes
15 years	S rank	46	558
4 years	A rank	34	415
1 year	B rank	34	415

learn a simultaneous ST system. This has the potential to allow our system to learn not only segmentation, but also rewordings such as those shown in Figure 1, or other tricks interpreters use to translate more efficiently.

In this work, we first collect simultaneous interpretation data from professional simultaneous interpreters of three levels of experience. Next, we use the simultaneous interpretation data for constructing a simultaneous ST system, examining the effects of using data from interpreters on the language model, translation model, and tuning. As a result, the constructed system has lower delay, and achieves translation results closer to a highly experienced simultaneous interpreter than when translation data alone is used in training. We also find that according to automatic evaluation metrics, our system achieves performance similar to that of a simultaneous interpreter that has 1 year of experience.

2. Simultaneous interpretation data

As the first step to performing our research, we first must collect simultaneous interpretation data. In this section, we describe how we did so with the cooperation of professional simultaneous interpreters. A fuller description of the corpus will be published in [8].

2.1. Materials

As materials for the simultaneous interpreters to translate, we used TED[1] talks, and had the interpreters translate in real time from English to Japanese while watching and listening to the TED videos. We have several reasons for using TED talks. The first is that for many of the TED talks there are already Japanese subtitles available. This makes it possible to compare data created by translators (i.e. the subtitles) with simultaneous interpretation data. TED is also an attractive testbed for machine translation systems, as it covers a wide variety of topics of interest to a wide variety of listeners. On the other hand, in discussions with the simultaneous interpreters, they also pointed out that the wide variety of topics and highly prepared and fluid speaking style makes it a particularly difficult target for simultaneous interpretation.

2.2. Interpreters

Three simultaneous interpreters cooperated with the recording. The profile of interpreters is shown in Table 1. The most important element of the interpreter's profile is the length of their experience as a professional simultaneous interpreter. Each rank is decided by the years of experience. By comparing data from simultaneous interpretation of each rank, it is likely that we will be able to collect a variety of data based on rank, particularly allowing us to compare better translation to those that are not as good. Note that all of the interpreters work as professionals and have a mother tongue of Japanese. The number of lectures interpreted is 34 lectures for the A and B ranked interpreters, and 46 lectures for the S rank interpreter.

[1] http://www.ted.com

```
0001 - 00:44:107 - 00:45:043
本日は<H>
0002 - 00:45:552 - 00:49:206
みなさまに(F え)難しい話題についてお話したいと思います。
0003 - 00:49:995 - 00:52:792
(F え)みなさんにとっても意外と身近な話題です。
```

Figure 2: Example of a transcript in Japanese with annotation for time, as well as tags for fillers (F) and disfluencies (H)

Table 2: Translation and simultaneous interpretation data

Data		Lines	Words(EN)	Words(JA)
Translation	T1	167	3.11k	4.58k
	T2			4.64k
Simultaneous interpretation	I1			4.44k
	I2			3.67k

2.3. Transcript

After recording the simultaneous interpretation, a transcript is made from the recorded data. An example of the transcript is shown in Figure 2. The utterance is divided into utterances using pauses of 0.5 seconds or more. The time information (e.g., start and end time of each utterance) and the linguistic information (e.g., fillers and disfluencies) are tagged.

3. Difference between translation data and simultaneous interpretation data

In this section, in order to examine the differences between data created using simultaneous interpretation and time-unconstrained translation, we compare the translation data with the simultaneous interpretation data.

3.1. Setup

To perform the comparison, we prepare two varieties of translation data, and two varieties of simultaneous interpretation data. The detail about the corpus is shown in Table 2. For the first variety of translation data (T1), we had an experienced translator translate the TED data from English to Japanese without time constraints. For the second variety of translation data (T2), we used the official TED subtitles,

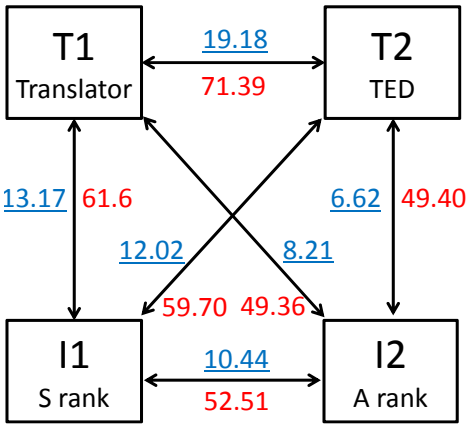

Figure 3: Results of similarity measurements between interpreters and translators. The underlined score is BLEU and the plain score is RIBES

generated and checked by voluntary translators. For the two varieties of interpretation data, I1 and I2, we used the transcriptions of the interpretations performed by the S rank and A rank interpreter respectively.

The first motivation for collecting this data is that it may allow us to quantitatively measure the similarity or difference between interpretations and translations automatically. In order to calculate the similarity between each of these pieces of data, we use the automatic similarity measures BLEU [9] and RIBES [10]. As BLEU and RIBES are not symmetric, we average BLEU or RIBES in both directions. For example, we calculate for BLEU using

$$\frac{1}{2}\{\mathrm{BLEU}(R,H)+\mathrm{BLEU}(H,R)\} \quad (1)$$

where R and H are the reference and the hypothesis. Based on this data, if the similarities of T1-T2 and I1-I2 are higher than T1-I1, T2-I1, T1-I2 and T2-I2, we can find that there are real differences between the output produced by translators and interpreters, more so than the superficial differences produced by varying expressions.

3.2. Result

The result of the similarity is shown in Figure 3. First, we focus on the relationship between the two varieties of translation data.

For T1-T2, BLEU is 19.18 and RIBES is 71.39, the highest of all in all combinations. Thus, we can say that the two translators are generating the most similar output. Next, we focus on the relationship between the translation and the simultaneous interpretation data. The similarity of T1-I1, T2-I1, T1-I2 and T2-I2 are all lower than T1-T2. In other words, interpreters are generating output that is significantly different from the translators, much more so than is explained by the variation between the translators themselves.

However, we see somewhat unexpected results when examining the relationship between the data from the two simultaneous interpreters. For I1-I2, BLEU is 10.44 and RIBES is 52.51, much lower than that of T1-T2. One of the reasons for this is the level of experience. From Table 2, we can see that the number of words translated by the A rank interpreter in I2 is almost 20 % less than that of the number of words translated by the S rank interpreter in I1. This is due to cases where the S rank interpreter can successfully interpret the content, but the A rank interpreter cannot. It is also notable that the S rank interpreter is translating almost as many words as the translation data, indicating that there is very little loss of content in the S rank interpreter's output.

However, it should be noted that I2 is more similar to I1 than either of the translators. Thus, from the view of the similarity measures used for automatic evaluation of translation, translation and simultaneous interpretation are different. Thus, in the following sections where we attempt to build a machine translation system that can generate output in a similar style to a simultaneous interpreter, we decide to evaluate our system against not the translation data, but the interpretation data of S1, which both manages to maintain the majority of the content, and is translating in the style of simultaneous interpreters.

4. Using simultaneous interpretation data

We investigate several ways of incorporating the data described in Section 2 into the MT training process.

4.1. Learning of the machine translation system

To attempt to learn a system that can generate translations similar to those of a simultaneous interpreter, we introduced simultaneous interpretation data into three steps of learning the MT system.

Tuning (Tu) : Tuning optimizes the parameters of models in statistical machine translation. The effect we hope to obtain by tuning towards simultaneous interpretation data is the learning of parameters that more closely match the translation style of simultaneous interpreters. For example, we could expect the translation system to learn to generate shorter, more concise translations, or favor translations with less reordering. In order to do so, we simply use simultaneous interpretation data instead of translation data for the development set used in tuning.

Language model (LM) : The LM has a large effect on word order and lexical choice of the translation result. We can thus assume that incorporating simultaneous interpretation data in the training of the LM will be effective to make translation results more similar to simultaneous interpretation. We create the LM using translation and interpretation data by making use of linear interpolation, with the interpolation coefficients

tuned on a development set of simultaneous interpretation data. This helps relieve problems of data sparsity that would occur if we only used simultaneous interpretation data in LM training.

Translation model (TM) : The TM, like the LM, also has a large effect on lexical choice, and thus we attempt to adapt it to simultaneous translation data as well. We adopt the phrase table by using the fill-up [11] method, which preserves all the entries and scores coming from the simultaneous interpretation phrase table, and adds entries and scores from the phrase table trained with translation data only if new.

4.2. Learning of translation timing

While in the previous section we proposed methods to mimic the word ordering of a simultaneous interpreter, our interpretation will not get any faster if we only start translating after each sentence finishes, regardless of word order. Thus, we also need a method to choose when we can begin translation mid-sentence.

In our experiment (Section 5), we use the method of Fujita et al. [4] to decide the translation timing according to each phrase's right probability (RP). This method was designed for simultaneous speech translation, and decides in real time whether or not to start translating based on a threshold for each phrase's RP, which shows the degree to which the order of the source and target language can be expected to be the same. For phrases where the RP is high, it is unlikely that a reordering will occur, and thus we can start translation, even mid-sentence, with a relatively low chance of damaging the final output. On the other hand, if an RP is low, starting translation of the phrase prematurely may cause un-natural word ordering in the output. Thus, Fujita et al. choose a threshold for the RP of each phrase, and when the current phrase at the end of the input has an RP that exceeds the threshold, translation is started, but when the current phrase is under the threshold, the system waits for more words before starting translation.

While Fujita et al. calculated their RPs from translation data, there is a possibility that interpreters will use less reordering than translators for many source language phrases. To take account of this, we simply make the RP table from translation data and simultaneous interpretation data. Using this method, we can hope that the system will be able to choose earlier timing to translate without a degradation in the translation accuracy. We calculate the RP from translation and interpretation data by simply concatenating the data before calculation.

5. Experiment

5.1. Data

In our experiment, the task is translating TED talks from English to Japanese. We use the translation and the interpretation data from TED as described in Section 2. As this data is still rather small to train a reasonably accurate machine translation system, we also use the EIJIRO dictionary and the accompanying example sentences[2] in our training data.

Table 3: The number of words in the data we used for learning translation model (TM), language model (LM), tuning (tune) and test set (test). The kinds of data are TED translation data (TED-T), TED simultaneous interpretation data (TED-I) and a dictionary with its corresponding example sentences (DICT)

	TED-T	TED-I	DICT
TM/LM (en)	1.57M	29.7k	13.2M
TM/LM (ja)	2.24M	33.9k	19.1M
tune (en)	12.9k	12.9k	—
tune (ja)	19.1k	16.1k	—
test (en)	—	11.5k	—
test (ja)	—	14.9k	—

The details of the corpus are shown in Table 3. As simultaneous interpretation data for both training and testing, we use the data from the S rank interpreter. This is because the S rank interpreter has the longest experience of the three simultaneous interpreters, and as shown empirically in Section 3, is able to translate significantly more content than the A rank interpreter. As it is necessary to create sentence alignments between the simultaneous interpretation data and TED subtitles, we use the Champollion toolkit [12] to create the alignments for the LM/TM training data, and manually align the sentences for the tuning and testing data.

5.2. Toolkit and evaluation method

As a machine translation engine, we use the Moses [13] phrase-based translation toolkit. The tokenization script in the Moses toolkit is used as an English tokenizer. KyTea [14] is used as a Japanese tokenizer. GIZA++ [15] is used for word alignment and SRILM [16] is used to train a Kneser-Ney smoothed 5-gram LM. Minimum Error Rate Training [17] is used for tuning to optimize BLEU. The distortion limit during decoding is set to 12, which gave the best accuracy on the development set.

The system is evaluated by the translation accuracy and the delay. BLEU [9] and RIBES [10] are used to calculate translation accuracy. RIBES is an evaluation method that focuses on word reordering information, and is known to work well for the language pairs that have very different grammatical structure like English-Japanese. The delay D is calculated as $D = U + T$. U is the average amount of time that we must wait before we can start translating, and T is the time required for MT decoding. Note that, in this experiment, we make the simplifying assumption that we have 100% accurate ASR that can recognize each word in exactly real time,

[2] Available from http://eijiro.jp

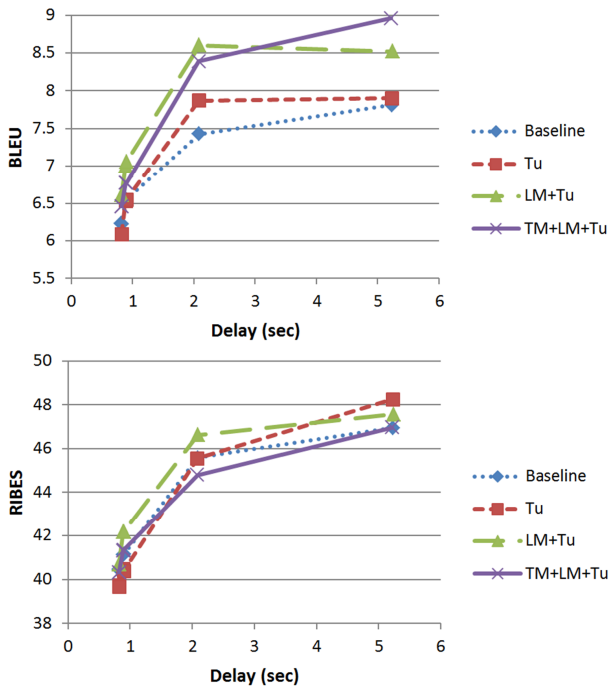

Figure 4: Result of machine translation system

and do not consider the time required for speech synthesis.

5.3. Result: Learning of the MT system

Simultaneous interpretation data is used in the three processes described in Section 4.1. To compare each variety of training, we experiment with 4 patterns:

Baseline: only translation data (w/o TED simultaneous interpretation data)

Tu: TED simultaneous interpretation data for tuning

LM+Tu: TED simultaneous interpretation data for LM training and tuning

TM+LM+Tu: TED simultaneous interpretation data for TM training, LM training and tuning

We decide the timing for translation according to the method described in Section 4.2, using a RP threshold of 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0.

The result of BLEU and delay is shown in the upper part of Figure 4. From these results, we can see that Tu does not show a significant improvement compared to the baseline, while LM+Tu and TM+LM+Tu show a significant improvement. For example, when the BLEU is 7.81[3], the delay is 5.23 seconds in the baseline, while in TM+LM+Tu the BLEU is 8.39, the delay is only 2.08 seconds. On the other hand, the result of RIBES and delay is shown in the lower part of Figure 4. In terms of RIBES, Tu, LM+Tu, and TM+LM+Tu do not show a significant improvement compared to the baseline. One of the reasons for this is tuning. When tuning, the parameters are optimized for BLEU, not RIBES. It should be noted that these numbers are all calculated using the S Rank interpreter's translations as a reference. In contrast, when we use the TED subtitles as a reference, the results for the baseline (BLEU=12.79, RIBES=55.36) were higher than those for TM+LM+Tu (BLEU=10.38, RIBES=53.94). From this experiment, we can see that by introducing simultaneous interpretation data in the training process of our machine translation system, we are able to create a system that produces output closer to that of a skilled simultaneous interpreter, although this may result in output that is further from that of time-unconstrained translators.

An example of results for the simultaneous interpreter, baseline, and TM+LM+Tu is shown in Table 4. From this example, we can see that the length of TM+LM+Tu is shorter than the baseline and is similar to the reference of simultaneous interpretation, as the length is adjusted during tuning. In this case, the reason for this is because the starting phrase in the baseline "見てみると" ("looking at") in baseline changes "では" ("ok") in TM+LM+Tu. Both translations are reasonable in this context, but the adapted system is able to choose the shorter one to reduce the number of words slightly. Another good example of how lexical choice was affected by adaptation to the simultaneous translations is the use of connectives between utterances. For example, the S rank simultaneous interpreter often connected two sentences by starting a sentence with the word "で" ("and"), likely to avoid long empty pauses while he was waiting for input. This was observed in 149 sentences out of 590 in the test set (over 25%). Our system was able to learn this distinct feature of simultaneous interpretation to some extent. In the baseline there were only 34 sentences starting with this word, while in TM+LM+Tu there were 81.

5.4. Result: Learning of translation timing

Next, we compare when the translation and the simultaneous interpretation data are used for learning of the RP (With TED-I) with when only translation data is used (W/O TED-I). The MT system is TM+LM+Tu for both settings.

The result is shown in Figure 5. From these two graphs, there is no difference in the translation accuracy and delay. We can hypothesize two reasons for this. First, the size of the simultaneous interpretation corpus is too small. The number of English words in the TED translation data is 1.57M, however, that in the TED simultaneous interpretation data is 29.7k. The second reason lies in the method we adopted for learning the RP table. In this experiment, the RP table is simply made by concatenating the translation data and simultaneous interpretation data. One potential way of solving this

[3]We speculate that the reason for these relatively low BLEU scores is the different grammatical structure between English and Japanese, and the highly stylized format of TED talks. Due to these factors, there is a lot of flexibility in choosing a translation, so the difference in lexical choice by translators might negatively affect the BLEU score.

Table 4: Example of translation results

	Sentence
Source	if you look at in the context of history you can see what this is doing
S Rank Reference	過去から / 流れを見てみますと / 災害は / このように / 増えています from the past / look at the context and / disasters are / like this / increasing
Baseline (RP 1.0)	見てみると / 歴史の中で / 見ることができます / これがやっていること looking at / in the history / you can see / what this is doing
TM+LM+Tu (RP 1.0)	では / 歴史の中で / 見ることができます / これがやっていること ok / in the history / you can see / what this is doing

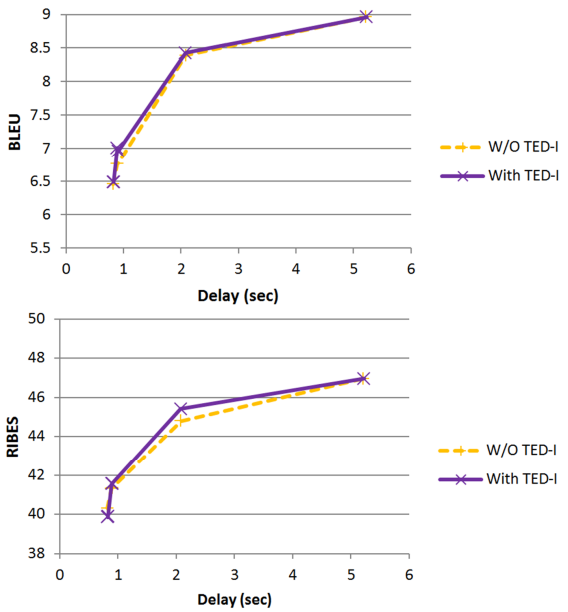

Figure 5: Result of dividing position

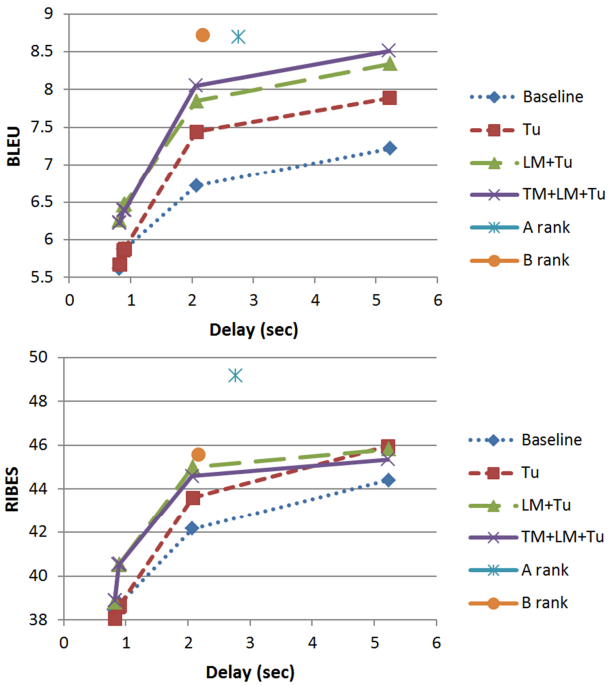

Figure 6: Result of comparing the system with human simultaneous interpreters

problem is, like we did for the TM, creating the table using the fill-up method.

5.5. Result: Comparing the system with human simultaneous interpreters

Finally, we compare the simultaneous ST system with human simultaneous interpreters. Simultaneous interpretation (and particularly that of material like TED talks) is a difficult task for humans, so it would be interesting to see how close are automatic systems are to achieving accuracy in comparison to imperfect humans. In the previous experiments, we assumed an ASR system that made no transcription errors, but if we are to compare with actual interpreters, this is an unfair comparison, as interpreters are also required to accurately listen to the speech before they translate. Thus, in this experiment, we use ASR results as input to the translation system. The word error rate is 19.36%. We show the results of our translation systems, as well as the A rank (4 years) and B rank (1 year) interpreters in Figure 6.

First, comparing the results of the automatic systems with Figure 4, we can see that the accuracy is slightly lower in terms of BLEU and RIBES. However the overall trend is almost same. From the view of BLEU, the system achieves results slightly lower than those of human simultaneous interpreters. However from the view of RIBES, the automatic system and B rank interpreter achieve similar results. So the performance of the system is similar, but likely slightly inferior to the B rank interpreter. It is also interesting to note the delay of the simultaneous interpreters. Around two seconds of delay is the shortest delay with which the system can translate while maintaining the translation quality. As well, the simultaneous interpreters begin to interpret two to three seconds after the utterance starts. We hypothesize that it is difficult to begin earlier than this timing while maintaining

the translation quality, both for humans and machines.

6. Conclusions

In this paper, we investigated the effects of constructing simultaneous ST system using simultaneous interpretation data for learning. As a result, we find the translation system grows closer to the translation style of a highly experienced professional interpreter. We also find that the translation accuracy has approached that of a simultaneous interpreter with 1 year of experience according to automatic evaluation measures. In the future, we are planning to do subjective evaluation, and analyze the differences in the style of translation between the systems in more detail.

7. Acknowledgments

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

8. References

[1] Roderick Jones. *Conference Interpreting Explained (Translation Practices Explained)*. St. Jerome Publishing, 2002.

[2] Koichiro Ryu, Atsushi Mizuno, Shigeki Matsubara, and Yasuyoshi Inagaki. Incremental Japanese spoken language generation in simultaneous machine interpretation. In *Proc. Asian Symposium on Natural Language Processing to Overcome language Barriers*, 2004.

[3] Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan Ladan Golipour, and Aura Jimenez. Real-time incremental speech-to-speech translation of dialogs. In *Proc. NAACL*, 2012.

[4] Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Simple, lexicalized choice of translation timing for simultaneous speech translation. In *Proc. 14th InterSpeech*, 2013.

[5] Matthias Paulik and Alex Waibel. Automatic translation from parallel speech: Simultaneous interpretation as mt training data. In *Proc. ASRU*, pages 496–501. IEEE, 2009.

[6] Vivek Kumar Rangarajan Sridhar, John Chen, and Srinivas Bangalore. Corpus analysis of simultaneous interpretation data for improving real time speech translation. In *Proceedings of InterSpeech*, 2013.

[7] Hitomi Toyama, Shigeki Matsubara, Koichiro Ryu, Nobuo Kawaguchi, and Yasuyoshi Inagaki. Ciair simultaneous interpretation corpus. In *Proc. Oriental COCOSDA*, 2004.

[8] Hiroaki Shimizu, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Collection of a simultaneous translation corpus for comparative analysis (in submission). In *Proc. LREC 2014*, 2014.

[9] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318, Philadelphia, USA, 2002.

[10] Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. Automatic evaluation of translation quality for distant language pairs. In *Proc. EMNLP*, pages 944–952, 2010.

[11] Arianna Bisazza, Nick Ruiz, and Marcello Federico. Fill-up versus interpolation methods for phrase-based smt adaptation. In *Proc. IWSLT*, pages 136–143, 2011.

[12] Xiaoyi Ma. Champollion: A robust parallel text sentence aligner. In *Proc. LREC*, 2006.

[13] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proc. ACL*, pages 177–180, Prague, Czech Republic, 2007.

[14] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proc. ACL*, pages 529–533, Portland, USA, June 2011.

[15] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[16] Andreas Stolcke. SRILM - an extensible language modeling toolkit. In *Proc. 7th International Conference on Speech and Language Processing (ICSLP)*, 2002.

[17] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proc. ACL*, 2003.