Simulating Segmentation by Simultaneous Interpreters for Simultaneous Machine Translation

Akiko Nakabayashi
The University of Tokyo
akinkbys@phiz.c.u-tokyo.ac.jp

Tsuneaki Kato
The University of Tokyo
kato@boz.c.u-tokyo.ac.jp

Abstract

One of the major issues in simultaneous machine translation setting is when to start translation. Inspired by the segmentation technique of human interpreters, we aim to simulate this technique for simultaneous machine translation. Using interpreters' output, we identify segment boundaries in source texts and use them to train a predictor of segment boundaries. Our experiment reveals that translation based on our approach achieves a better RIBES score than conventional sentence-level translation.

1 Introduction

Simultaneous interpreters listen to speech in a source language, translate it into a target language, and deliver the translation simultaneously. In this process, it is important to minimize the burden on the interpreters and to keep up with the original speech, particularly between language pairs with different word orders, such as English and Japanese. One of the tactics often used by interpreters is segmentation, which is to split a sentence according to "units of meaning" into multiple segments and translate those segments in sequence. Reformulation, simplification, and omission are further applied to generate natural translation (Jones, 1998; He et al., 2016).

In simultaneous machine translation, where speeches and lectures are translated simultaneously, this segmentation technique is also effective for minimizing translation latency. If translation is generated per sentence, as in a standard machine-translation process, there is a substantial delay between the original speech and its translation. By contrast, if a sentence is segmented into excessively short pieces, it becomes difficult to produce a meaningful translation from snippets of information. It is therefore important to determine the appropriate translation segment length that defines the correct timing to start the translation.

This study aims to learn the segmentation technique—which is necessary to realize fluent simultaneous machine translation with low latency—from human interpreters by examining their output, and to analyze this technique and propose a method to simulate it. The problem, however, is that the segments identified by interpreters are not always self-evident. Segments are considered to be produced by interpreters by splitting a sentence into "units of meaning," which is a minimal unit for interpreters to process information, but the linguistic characteristics of these "units of meaning" remain unclear. After discussing the background in Section 2, we return to present this issue in Section 3, where we propose an approach to identify segment boundaries in source texts using interpreters' output. Then, we demonstrate that segments identified by the proposed approach are plausible and that this segmentation approach can produce fluent translation with low latency. In Section 4, we describe the analysis of segment boundaries in the source texts. This analysis aims at understanding what factors determine those boundaries, as source texts are the only available means of identifying them in actual simultaneous-machine-translation settings. The result reveals that

33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33), pages 165-173, Hakodate, Japan, September 13-15, 2019

an intricate set of linguistic factors define segment boundaries. Based on this analysis, we propose a framework to simulate the segmentation technique of interpreters using a predictor based on a Recurrent Neural Network (RNN) and analyze the result in Section 5. Although this predictor does not reproduce interpreters' segmentation perfectly, the translation generated per predicted segment achieves better RIBES scores[1] (Isozaki et al., 2010) compared with conventional sentence-level translation.

Overall, the contributions of this study are

- We analyze the segmentation tactic of human interpreters and clarify what linguistic factors define segment boundaries.
- We propose a framework to simulate this segmentation. Our experiment reveals that this approach of learning segmentation tactic from human interpreters benefits simultaneous machine translation.

2 Background

The process of simultaneous interpreting involves segmentation and reformulation. How sentences are segmented defines the appropriate timing to start the translation, and identifying these timings is one of the major issues in the research on simultaneous machine translation. Various strategies have been explored to address this issue. Fügen et al. (2007) and Bangalore et al. (2012) tried to find clues based on linguistic and a non-linguistic features (such as commas and pauses) in the original speech, whereas segments defined assuming that the segmentation depends solely on a specific syntactical feature of a source text may not provide the best timing for the target text. Fujita et al. (2013) suggested determining the translation timings by referring to a translation phrase table. Oda et al. (2015) built a classifier to find segment boundaries that maximized the sum of translation quality indices. Recent studies focused more on end-to-end approaches by training translation timings and translation models all together to produce translations with high translation scores (Cho and Esipova, 2016; Ma et al., 2018). Among them, Cho and Esipova (2016), rather than segmenting the original speech before the translation process, chose to translate the original text incrementally and define the appropriate timing to fix the translation based on a prescribed criterion. In those approaches, sentence-level translation corpora were used to train and evaluate the models, which did not necessarily produce good results in simultaneous-machine-translation settings. However, interpreting corpora are limited in size; therefore, it is not realistic to use them in these approaches.

In this study, we propose utilizing a simultaneous-interpreting corpus to find segment boundaries on source texts and building a segmentation predictor based on them. As transcripts of simultaneous interpreters provide insight on how they split sentences into segments at appropriate timings, simulating this tactic and translating based on those segments are expected to yield translation close to actual simultaneous interpreting. Shimizu et al. (2014) also referred to interpreters' output to identify segment boundaries, but used a non-linguistic feature to find patterns of segmentation. We, by contrast, utilize linguistic features that appear in interpreters' output not only to identify segment boundaries, but also to predict them. Tohyama and Matsubara (2006) and He et al. (2016) conducted descriptive studies on the tactics of simultaneous interpreters.

After using a model to split sentences into segments, we assume that each segment is translated independently using a conventional translation model and that reformulation is applied if the segment is syntactically incomplete. This process produces the final translation output.

For analysis and experiment, we used CIAIR Simultaneous Interpreting Corpus (Toyama et al., 2004), which contains transcripts of interpreters who simultaneously interpreted monologues and conversation. Regarding monologues, there are 136 simultaneous-interpretation transcripts with 5,011 utterances for 50 English speeches with 2,849 utterances. We used 24 speeches recorded in 2000, which have the transcripts of four interpreters each.

[1] We use RIBES scores as our evaluation criteria because it factors in word order, which is critical in simultaneous interpreting.

3 Segment Boundaries in Simultaneous Interpreting

In this section, we address our approach to identifying segment boundaries, which focuses on the interpreting results. After describing our motivation, we explain our method and demonstrate its effectiveness.

3.1 What are Segment Boundaries?

Interpreters split a sentence according to "units of meaning" into segments and translate them in sequence, but what is a "unit of meaning?" Is it defined by speakers' pauses, or by punctuations, as previous works suggest? A "unit of meaning" cannot be systematically related to grammatical categories, and it changes depending on the speaker's utterance speed and languages pairs (Jones, 1998). Based on the idea that a "unit of meaning" is a cognitive representation in the listener's mind (Jones, 1998), we see that the segment boundaries identified by interpreters appear in their output. The following example shows that it is difficult to identify segment boundaries by solely examining a source text, whereas the interpretation output provides some clues about the segments recognized by the interpreters.

Source text:
If you do that, the ups and downs seem to level out and you build more. It's a natural way of making money. (SXPSX006.NX02.ETRANS)
Interpretation transcript:
その間にはいろいろな上下があると思いますがその長い期間の間にそういったものは平均が取れて最終的にはお金が儲かっていくと思います。(SXPSX006.L.IA08.JTRANS)
"During that time, there will be various ups and downs, I think, but during that long period, such things will be leveled out, and at the end, we can make money, I think."

The source text seems to suggest that it can be syntactically split between the conditional clause and the main clause in the first sentence, and between the first and the second sentence. However, when we look at the interpreter's transcripts, we can see that the interpreter split the sentence immediately after "いろいろな上下があると思いますが (there will be various ups and downs, I think)," which corresponds to the *the ups and downs* in the source text.

Jones (1998) further claimed that interpreters can start the translation "once they have enough material from the speaker to finish their own (interpreted) sentence." Given that sentences tend to become long with coordinate conjunctions connecting multiple clauses, and sentence boundaries are not clear in a spoken language, a sentence can be rephrased as a clause. In light of this idea, we believe that once interpreters identify a "unit of meaning',' they translate it and produce a clause. In other words, a clause in interpreters' output is the translation of a "unit of meaning" that they recognize. Therefore, we propose the following approach to identifying segment boundaries:

- Split interpreting results into clauses
- Identify segments on source speech texts by finding corresponding word strings

Clauses in the interpreters' output and the corresponding segments in the source speech texts were annotated manually in this study; however, we believe that these processes can be automated.

In the previous example, segment boundaries are considered to appear at the following positions in the source text.

Source text:
If you do that, the ups and downs seem / to level out / and you build more. It's a natural way of making money. /

3.2 Identified Segment Boundaries

We identified segment boundaries based on the aforementioned approach. We examined the segmentation distribution in the transcripts of four interpreters associated with the speech text file (SXPSX005.NX02.ETRANS). As shown in Table 1, the cumulative total of segment boundaries identified by four interpreters was 441 with 153 distinct places. All four interpreters agree to split segments at 64 distinct places, i.e., 256 places in total. Three out of four interpreters agree to split segments at 36 distinct places, i.e., 108 places in total. The cumulative total of segments on which three or more inter-

preters agree was 364 (82.5%) out of 441 segments. We believe that this number is sufficiently large to confirm that segment identification by the aforementioned approach is objective and usable. For further research, we extracted segment boundaries shared by three or four interpreters.

Number of Interpreters Agreed	Number of Segments (Total)
4	64 (256)
3	36 (108)
2	24 (48)
1	29 (29)
Total	153 (441)

Table 1: Distribution of segments (File SX-PSX005.NX02.ETRANS)

Table 2 shows an overview of the number of sentences and segments in 10 files. The first two lines show the total word count and total time spent for the 10 speeches. The total number of sentences in 10 files, average word count per sentence, average time spent per sentence, and those for segments follow.

Speech	Word Count	12,859
	Total Time (min:sec)	101:49
Sentence	Number of Sentences	510
	Average Word Count	25.2
	Average Time (min:sec)	0:12
Segment	Number of Segments	1,127
	Average Word Count	11.4
	Average Time (min:sec)	0:05

Table 2: Overview of sentences and segments

The average word count in a sentence is 25, while that in a segment is 11. Given that the average time spent on a segment is 5 seconds, and that spent on a sentence is 12 seconds, segment-level translation is considered to reduce translation latency by 7 seconds compared with sentence-level translation. This shows that the proposed segmentation approach contributes to reducing translation latency.

We compared the RIBES scores of the segment-level translation with those of the interpreters' transcript to prove that the translation generated by the proposed approach resembles the interpreters' output. After segment boundaries were identified, each segment was translated using Google Translate[2]. The translated segments were concatenated and used as final translation. Reformulation was not applied in this experiment. The RIBES score of this segment-level translation was 0.7755, with the transcripts of three interpreters used as the reference translation. The RIBES score of the other interpreter's transcript was 0.7754 and that of the sentence-level translation was 0.7412 using with the same reference translation.

The fact that the RIBES score of the segment-level translation with the proposed approach was close to that of an interpretation transcript suggests that the proposed approach can generate translation comparable to interpreters' output and that considering a clause in such output to be a "unit of meaning" is plausible and realistic.

4 Characteristics of Segments

While we identified segments through a relationship with interpreters' output as in the previous section, the interpreters' output is not available in actual simultaneous-machine-translation settings when predicting segment boundaries. We analyzed the segments and their characteristics to understand what factors determine the segment boundaries in the source texts.

We extracted part-of-speech (POS) bigrams before and after the segment boundaries to determine where such boundaries tend to appear. Table 3 shows the top six patterns of segment boundaries. The numbers in parentheses show the proportion of segment boundaries to the places where each pattern appears.

Feature	Number of Segment Boundaries
After "."	1,207 (98 .3%)
Before Coordinate Conjunction	927 (55.7%)
After ","	545 (39.5%)
Before Wh	114 (20.3%)
Before Adverb	240 (11.3%)
Before Preposition/ Subordinate Conjunction	377 (10.8%)

Table 3: Characteristics of segment boundaries

While periods are a strong indication for defin-

[2]https://translate.google.com [Accessed: 9 Jan, 2019]

ing segment boundaries, they also appear elsewhere, such as before coordinate conjunctions and after commas[3]. However, not all positions with these features become segment boundaries, and various linguistic factors other than POS also seem to play an important role in the decision. For example, a conjunction may coordinate noun phrases or clauses. If the conjunction coordinates clauses, it is more likely that word strings before the conjunction become a segment than when it coordinates noun phrases. However, this information cannot be captured by examining POS n-grams alone.

The last five bigram patterns are discriminative in simultaneous interpreting (Tohyama and Matsubara, 2006). We focused on relative pronouns, which include wh-determiners and subordinate conjunctions, and further analyzed what factors influence decisions on whether a relative clause with a relative pronoun becomes a segment or not. In Japanese, a relative clause comes before the antecedent, and sentence-level translation usually employs this word order. However, in simultaneous interpreting, the antecedent is often translated before the relative clause is fully uttered.

We built a logistic regression classifier to predict segment boundaries and investigated the weight of each feature to determine what factors contribute to the decision on segmentation. The features used in the classification model were: the number of words in the relative clause, the syntactic role of the antecedent in the main clause, the syntactic role of the relative pronoun in the relative clause, and the presence of comma before the relative pronoun. Concerning the syntactic role of an antecedent in the main clause and that of a relative pronoun in the relative clause, if the antecedent or the relative pronoun appeared before the verb, we assumed its syntactic roles to be "subject (SBJ)"; otherwise, it was "object (OBJ)." The values of the features were normalized before training the logistic regression. We used the NLTK[4] package to predict the POS and the scikit-learn[5] package to build the logistic regression models. A total of 45 POS, including symbols, were defined in NLTK. The accuracy of this model was 0.687, although classification was not the primary objective.

Table 4 shows the weight of each feature. A large absolute value of the weight means high contribution of the feature to the decision on segmentation, and a large positive value of the weight means that the position with the feature is likely to become a segmentation boundary.

Feature	Weight	p-value
Number of Words in Relative Clause	0.3048	$<$0.001
Role in Main Clause (SBJ)	-0.4502	$<$0.001
Role in Relative Clause (SBJ)	0.1169	0.11
Comma	0.3132	$<$0.001

Table 4: Factors influencing segment boundaries

This result shows that the number of words in the relative clause, the role of the antecedent in the main clause, and the presence of a comma are within the level of significance and are important for the definition of segment boundaries. Rather than a single factor, multiple linguistic factors contribute to the decision of simultaneous interpreters on where to split a sentence.

5 Predicting Segment Boundaries

In this section, we describe our segmentation framework for simultaneous machine translation. The data presented in the previous section show that clues on segmentation cannot be explained by a single feature. To integrate such intricate features, we built an RNN-based predictor of segment boundaries. After explaining its architecture, we show the results of experiments performed to examine whether it can capture these linguistic features and simulate interpreters' tactics.

It is worth noting that in simultaneous-machine-translation settings, words in the source text become available one by one, and the entire sentence is not available to be parsed as in the analysis presented in Section 4. Hence, all the linguistic features stated in Section 4 may not be readily available when predicting segment boundaries. We expect that those features are somehow represented in and related to the already available context. The predictor predicts

[3]Commas and periods are annotated in the transcriptions. We believe corresponding information can be captured through acoustic information in actual simultaneous-machine-translation settings.

[4]http://www.nltk.org

[5]https://scikit-learn.org/stable/

segment boundaries using segmented source texts generated by the approach proposed in Section 3 as training data. It was modeled as a binary classification with the task of predicting whether a segment boundary appears in front of an input word, when a word sequence is given as input.

5.1 Experiment Settings

The model uses the long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) of an RNN. We use a uni-directional RNN, given that the whole sentence is not available when predicting segment boundaries and words in the source texts become available one by one in sequence in simultaneous-machine-translation settings. Figure 1 shows an overview of the model.

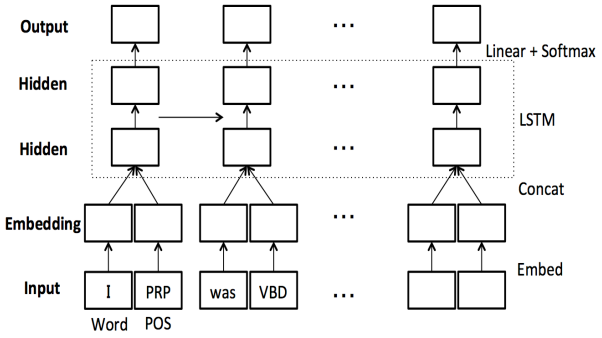

Figure 1: Model for predicting segment boundaries

Each word and its POS in the input layer are represented as one-hot vectors. The word and POS are concatenated after being mapped to the embedding layer with a lower dimension; this concatenated embedding is then used as the input for the next LSTM layer. The output of the LSTM is mapped to a two-dimensional vector and the result of applying the softmax function to the vector shows the probability of the input being assigned to each class. We used cross-entropy as a loss function.

The Chainer[6] package is used to implement the model. The dimension of word embeddings is set to 300, while that of POS embeddings is 4. The input and the output of the LSTM have 304 dimensions. The dropout rate is set to 0.1 and the class weight is set to 3 to deal with biased samples.

[6]https://chainer.org

5.2 Training Data and Test Data

Out of 24 speeches, 22 were used as a training dataset, one as a development set, and one as a test set. Table 5 provides an overview of the data used.

	Training Data	Test Data
Tokens	30,151	1,197 (OOV: 231)
Types	3,278	410
Sentences	1,097	70
Segment Boundaries	2,413	130
Non Segment Boundaries	27,738	1,067

Table 5: Size of data

The numbers of unknown words that appear in the test set but not in the training set (Out-of-vocabulary; OOV) are large.

Words on segment boundaries constitute only 8.0% of the total number of words, so the number of words in each class is biased.

Each datum is labeled as described below. Training and testing are executed per sentence. Class "1" shows that a segment boundary comes before the corresponding word.

Words: "I", "was", "traveling", "in", "Europe", "and", "when", "I", "was", "in", "Greece", ",", "I", "met", "a", "man", "from", "Holland", "."
Label: (1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)

5.3 Experiment Results

Table 6 shows the precision, recall, and F-measure of the experiment. The F-measure for class "1" was 0. 80.

Class	Precision	Recall	F-measure
1	0.82	0.78	0.80
0	0.97	0.98	0.98

Table 6: Results of segment boundary prediction

We further analyzed the prediction results for the comma, coordinate conjunctions, and wh-clauses by twelve-fold cross-validation. Often, but not always, interpreters split sentences before those words in simultaneous interpreting, and various factors are considered to influence their decisions as we discussed

in the previous section. Table 7 shows the baseline, accuracy, recall, and precision for the comma, coordinate conjunctions, and wh-clauses. We used the most common prediction, which is the probability of predicting the biggest category by chance, as our baseline.

	Before Coordinate Conjunction	After ","	Before Wh-clause
Baseline	55.7	60.5	79.7
Accuracy	66.0	66.2	80.9
Precision	66.1	55.1	52.5
Recall	80.3	76.7	64.9

Table 7: Results for coordinate conjunctions, comma, and wh-clauses

The accuracy for all three cases was higher than baseline, and we can say that the model could capture more information than POS has. To further investigate the results, we picked up two relative pronouns, *which* and *that*, to see if there are any significant differences between words. Table 8 shows the number of boundaries, as well as the baseline, accuracy, recall, and precision for *which* and *that*.

	which	that
Total Number	69	116
Number of Boundaries	33	16
Baseline	52.2	86.2
Accuracy	68.1	75.0
Precision	63.4	15.8
Recall	78.8	18.8

Table 8: Results for relative pronouns, *which* and *that*

Segment boundaries often appear before *which*, while they do not before *that*. The accuracy for the relative pronoun *which* was higher than baseline. The following example shows a case where the model correctly predicted the segment boundary before the relative pronoun *which*.

Label: People are a little more casual, / they take their time / and they're really very friendly / which is something that makes me feel a lot better. / (SXUSX012)

Predicted: People are a little more casual, they take their time / and they're really very friendly / which is something that makes me feel a lot better. /

By contrast, the accuracy for *that* was lower than baseline. This may be attributed to the small positive training dataset available for this relative pronoun.

5.4 Translation Results

After segmenting sentences at the predicted segment boundaries, we translated each segment using Google Translate. Then, we concatenated the translation outputs, and analyzed it by calculating their RIBES scores. The transcripts of four interpreters were used as the reference translation. Although reformation should be applied separately before yielding the final output, this analysis was conducted on the translation texts without reformation.

The RIBES score of segment translation with predicted segment boundaries was 0.7683, which is higher than the score of sentence translation, i.e., 0.7610. The RIBES score of segment translation with correct segment boundaries was 0.7964. Table 9 shows some examples. Translation results generated by the proposed approach had a word order similar to that of the interpreters' transcripts. By applying reformation, translation outputs are expected to become more natural.

Table 10 shows an example with a low RIBES score for segment translation with the predicted segment boundaries. The predictor failed to split the sentence before the preposition, splitting it at a wrong position, which caused a reduced RIBES score. These issues can be resolved by improving the accuracy of the segmentation.

6 Conclusion and Further Study

Segmentation is one of the key issues in the area of simultaneous machine translation. To resolve it, we proposed a method that uses interpreters' output. Specifically, we assumed that a "unit of meaning" appears as a clause in interpreters' output and identified segment boundaries by marking the corresponding position in the source texts. We analyzed them in the source texts and pointed out that various linguistic factors determine those boundaries.

We used segment boundaries in the source texts as training data to build a segmentation predictor that reproduces interpreters' segmentation strategies. The F-measure of the segmentation predictor

Segment Boundaries	The bagpipe is most commonly heard at highland games / where many bands gather to play music / and perform Scottish games such as the caber toss or the hammer throw .
Translation based on Segment Boundaries	バグパイプはハイランドゲームでよく聞かれます たくさんのバンドが集まって音楽を演奏する そして、キャバートスやハンマースローなどのスコットランドの試合を行います。 “The bagpipe is most commonly heard at highland games. Many bands gather and play music. And perform Scottish games, such as the caber toss or the hammer throw.”
Predicted Segment Boundaries	The bagpipe is most commonly heard at highland games / where many bands gather to play music / and perform Scottish games such as the caber toss / or the hammer throw.
Translation based on Predicted Segment Boundaries	バグパイプはハイランド ゲームでよく聞かれます 音楽を演奏するために多くのバンドが集まる場所 キャバートスなどのスコットランドのゲームを実行する またはハンマー投げ “The bagpipe is most commonly heard at highland games. The place many bands gather to play music. Perform Scottish games, such as the caber toss. Or the hammer throw.”
Interpreting Transcript	バグパイプが通常聞かれるのはハイランドゲームでです。そこで多くのバンドが集まりましてそして音楽を演奏します。そしてスコットランドのゲーム例えばケーバートスあるいはハンマースローなどを行います。 “The bagpipe is most commonly heard at highland games. There, many bands gather and play music. And perform Scottish games, such as the caber toss or the hammer throw.”

Table 9: Translation results

Segment Boundaries	This happened again and again / until several people were, of course, killed.
Translation based on Segment Boundaries	これは何度も何度も起こった もちろん数人が殺されるまで。 “This happened again and again. Of course until several people were killed.”
Predicted Segment Boundaries	This happened again and again until several people were, / of course, killed.
Translation based on Predicted Segment Boundaries	これは何人かの人々がなるまで何度も何度も起こりました、もちろん、殺した。 “This happened again and again until several people became. Of course killed.”
Interpreting Transcript	これが何度も何度も繰り返されました 。何人かの人が撃ち殺されるまでやったわけです。 “This happened again and again. Did it until several people were killed.”

Table 10: Translation results with low RIBES scores

was 0.80. Interpreters often split sentences before relative pronouns, and in many cases the predictor could predict segment boundaries correctly at such positions. When we split sentences at the predicted positions and translated each segment using Google Translate, the output had a word order similar to that of the interpreters' transcripts and its RIBES score was higher than that of sentence-level translation. This underscores that the proposed approach benefits simultaneous machine translation. However, incorrectly predicted segment boundaries degraded the translation quality. Therefore, further improvement in the accuracy of the segmentation is required. The reformation model is another topic for further study.

References

Bangalore, S., Rangarajan Sridhar, K. V., Kolan, P., Golipour, L., and Jimenez, A. 2012. Real-time Incremental Speech-to-Speech Translation of Dialogs. *Pro-*

ceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: 437-445.

Cho, K., and Esipova, M. 2016. Can neural machine translation do simultaneous translation? *arXiv:1606.02012*.

Fügen, C., Waibel, A., and Kolss, M. 2007. Simultaneous translation of lectures and speeches. *Machine translation*, 21(4): 209-252.

Fujita, T., Neubig, G., Sakti, S., Toda, T., and Nakamura, S. 2013. Simple, lexicalized choice of translation timing for simultaneous speech translation. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*: 3487-3491.

He, H., Boyd-Graber, J., and Daume III, H. 2016. Interpretese vs. Translationese: The Uniqueness of Human Strategies in Simultaneous Interpretation. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*: 971-976.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*9(8): 1735-1780.

Isozaki, H., Hirao, T., Duh, K., Sudoh, K., and Tsukada, H. 2010. Automatic Evaluation of Translation Quality for Distant Language Pairs. *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*: 944-952. Association for Computational Linguistics.

Jones, R. 1998. *Conference interpreting explained*. Routledge.

Shimizu, H., Neubig, G., Sakti, S., Toda, T., and Nakamura, S. 2014. Doujitsuuyaku Deta wo Riyou Shita Onsei Doujitsuuyaku no tameno Yakushutsu Taimingu Kettei Shuhou (A Method to Decide Translation Timing for Simultaneous Speech Translation Using Simultaneous Interpreting Data). *Proceedings of the Association for Natural Language Processing*: 294-297.

Ma, M., Huang, L., Xiong, H., Liu, K., Zhang, C., He, Z., Liu, H., Li, X., and Wang, H. 2018. Stacl: Simultaneous translation with integrated anticipation and controllable latency. *arXiv:1810.08398*.

Oda, Y., Neubig, G., Sakti, S., Toda, T., and Nakamura, S. 2015. Syntax-based Simultaneous Translation through Prediction of Unseen Syntactic Constituents. *Proceedings of the 53rd ACL*(1): 198-207.

Toyama, H., Matsubara, S., Ryu, K., Kawaguchi, N., and Inagaki, Y. 2004. CIAIR Simultaneous Interpretation Corpus. *Proceedings of the oriental chapter of the International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques for Speech Input/Output (Oriental COCOSDA 2004)*.

Tohyama, H., Matsubara, S. 2006. Collection of Simultaneous Interpreting Patterns by Using Bilingual Spoken Monologue Corpus. *International Conference on Language Resources and Evaluation (LREC2006)*: 2564-2569.