An Efficient and Effective Online Sentence Segmenter for Simultaneous Interpretation

Xiaolin Wang Andrew Finch Masao Utiyama Eiichiro Sumita
Advanced Translation Research and Development Promotion Center
National Institute of Information and Communications Technology, Japan
{xiaolin.wang,andrew.finch,mutiyama,eiichiro.sumita}@nict.go.jp

Abstract

Simultaneous interpretation is a very challenging application of machine translation in which the input is a stream of words from a speech recognition engine. The key problem is how to segment the stream in an online manner into units suitable for translation. The segmentation process proceeds by calculating a confidence score for each word that indicates the soundness of placing a sentence boundary after it, and then heuristics are employed to determine the position of the boundaries. Multiple variants of the confidence scoring method and segmentation heuristics were studied. Experimental results show that the best performing strategy is not only efficient in terms of average latency per word, but also achieved end-to-end translation quality close to an offline baseline, and close to oracle segmentation.

1 Introduction

Simultaneous interpretation performs spoken language translation in a online manner. A spoken language translation system automatically translates text from an automatic speech recognition (ASR) system into another language. Spoken language translation itself is an important application of machine translation (MT) because it takes one of the most natural forms of human communication – speech – as input (Peitz et al., 2011). Simultaneous interpretation is even more demanding than spoken language translation because the processing must occur online.

Simultaneous interpretation can bridge the language gap in people's daily lives transparently because of its ability to respond immediately to users' speech input. Simultaneous interpretation systems recognize and translate speech at the same time the speakers are speaking, thus the audience can hear the translation and catch the meaning without delay. Potential applications of simultaneous interpretation include interpreting speeches and supporting cross-lingual conversation.

This paper is devoted to online sentence segmentation methods for simultaneous interpretation. Simultaneous interpretation systems are normally comprised of ASR systems and MT systems. The output of ASR systems is typically streams of words, but the input to MT systems is normally sentences. Sentence segmenters bridge this gap by segmenting stream of words into sentences. Figure 1 illustrates this process.

A number of segmentation methods have been proposed to pipeline ASR and MT, yet most of them require a long context of future words that follow sentence boundaries. In addition, they are often computationally expensive. These shortages make them unattractive for use in simultaneous interpretation. To the best of our knowledge, there are no published ready-to-use online sentence segmenters, and this motivated this paper. The proposed method is crafted in a way that requires little computation and minimum future words in order to achieve efficiency. Also the proposed method is directly optimized against the widely used measurement of translation quality – BLEU (Papineni et al., 2002) – in order to achieve effectiveness. We believe that this work can directly contribute to the development of real-world simultaneous interpretation systems.

The main contributions of this paper are,

- proposing a segment boundary confidence score;

Proceedings of the 3rd Workshop on Asian Translation,
pages 139–148, Osaka, Japan, December 11-17 2016.

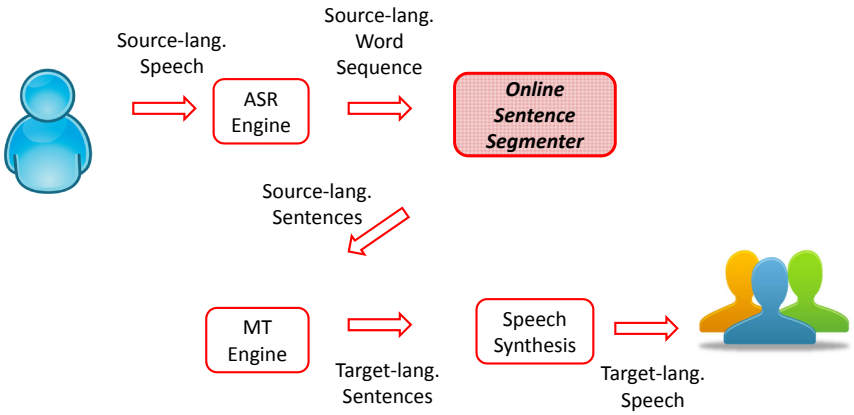

Figure 1: Illustration of Online Sentence Segmenter in Simultaneous Interpretation System

- proposing a hybrid online sentence segmenter;
- an empirical study and analysis of the proposed method on two translation tasks.

The rest of this paper is organized as follows. Section 2 reviews related works on segmentation methods. Section 3 describes our methods. Section 4 presents experiments between English and Japanese. Section 5 concludes this paper with a description of future work.

2 Related Works

A number of methods have been proposed to segment the output of ASR for MT. The works of Stolcke and Shriberg (1996), and Stolcke et al. (1998) are most related to this paper. They treated segmentation boundaries as hidden events occurring between words. They used N-gram language models and Viterbi algorithms to find the most likely sequences of hidden events. We admire them for approaching the problem by language models, which are well-studied techniques that run very fast. For sentence segmentation, they can tackle the task through the insertion of hidden beginning and end of sentence events. However, Stolcke et al. employed off-line Viterbi algorithms that require a long context of words. This will causes long latency for simultaneous interpretation. Therefore, this work has focused on developing lower-latency segmenters that require only one future word. Please note that Stolcke et al.'s methods are implemented using the SRILM toolkit, and it is used as a baseline (denoted *Hidden N-gram*) in our experiments.

Fügen et al. (2007) and Bangalore et al. (2012) proposed using pauses captured by ASR to denote segmentation boundaries. However, studies on human interpreters show that segmenting merely by pauses is insufficient, as human speakers may not pause between sentences. The mean proportion of silence-based chunking by interpreters is 6.6% when the source is English, 10% when it is French, and 17.1% when it is German (Venuti, 2012). Therefore, this paper focuses on using linguistic information. Nevertheless, pauses can be directly integrated into the proposed segment boundary confidence scores to boost performance.

Matusov et al. (2006) proposed a sentence segmentation algorithm which is similar to a conditional random field (CRF) (Lafferty et al., 2001). Lu and Ng (2010) applied CRFs to punctuation prediction which is an almost equivalent task to sentence segmentation. These CRF-based methods achieve high performance as they are able to integrate arbitrary features. However, CRF models take whole sequences as input, thus they cannot be directly applied in an online manner. Online CRFs are beyond the scope of this paper, and we plan to explore this in the future.

Ha et al. (2015) approached sentence segmentation by training a specialized monolingual machine translation system. Kzai et al. (2015) proposed a neural network approach to sentence segmentation. These two methods both require whole sequences as input, and require heavy computation. Therefore, they might not be suitable for online segmentation.

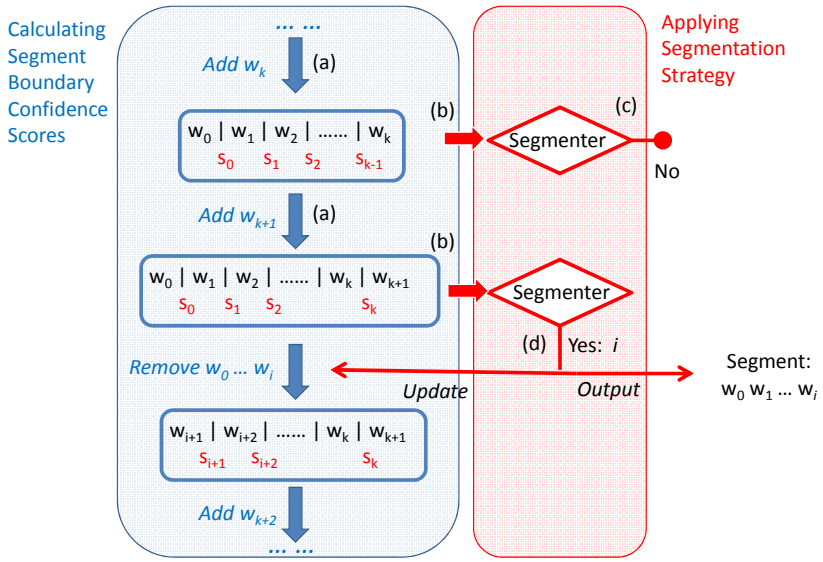

Figure 2: Illustration of Proposed Online Sentence Segmenter

A number of segmentation strategies targeted at splitting an input sentence into smaller pieces for simultaneous interpretation. Yarmohammadi et al. (2013) extracted monotonic phrase alignments from word-aligned sentence pairs and assumed that these phrasal alignments segment the source sentences in an appropriate manner for MT. They used the segmented data to train a binary classifier to predict segmentation boundaries.

Oda et al. (2014) built training data sets for segmentation through a greedy search algorithm, which searches for segmentation boundaries that yield high BLEU score. They trained a linear SVM (Cortes and Vapnik, 1995) to predict these boundaries. To mitigate the effect of noise in the training data sets, they further introduced feature grouping and dynamic programming to the raw greedy search algorithm.

Fujita et al. (2013) used phrase tables and reordering probabilities in phrase-based translation systems to segment an input sentence. Two heuristics were used for segmentation: in the first, if the partial input doesn't exist in phrase tables, segmentation boundaries are generated; in the second, if the right probability of reordering is less than a predefined threshold, segmentation boundaries are generated.

These works aim at outputting shorter segments than sentences, [1] which is capable of further reducing the latency in simultaneous interpretation. However, they assumed that input stream is already segmented into sentences, which is the topic of this paper. As such, our method is orthogonal to these methods, and it would be possible to pipeline our proposed method with them; we plan to explore this in the future. Another shortcoming of these works is that they are tied to specific translation systems, and this narrows their applicability.

3 Methodology

The proposed online sentence segmenters have two components – boundary confidence scoring and segmentation strategies (illustrated in Figure 2 and Algorithm 1). The input is a stream of words, denoted as $w_0, \cdots, w_i, w_{i+1}, \cdots, w_{k+1}$. The boundary confidence score, denoted as s_i, indicates the fitness of breaking after the i-th word. Segment strategies decide whether or not break based on confidence scores, denoted as b_i. The final output is a segmented sentence, e.g. $w_0, \cdots, w_i$.

The proposed segmenters work in an online manner as follows: words are input one by one. The sequence of input words and the derived confidence scores are maintained as states. Once an word is input, its confidence score is calculated and added into the sequence (which is labeled as a in Figure 2). Then a segmentation strategy is applied to the sequence (labeled as (b) in Figure 2). In case that the

[1] Fujita et al. (2013)'s method may work on word streams without sentence boundaries; Oda et al. (2014)s' segmentation model uses linear SVMs and local features extracted from just three word lookahead, so it might be adapted.

Algorithm 1 Online Sentence Segmenter

Require: $w_0, w_1, w_2, \ldots,$
$W \leftarrow []$; $S \leftarrow []$
for w_k in stream of words **do**
  $W \leftarrow W + [w_k]$ ▷ assume $W = [w_0, w_1, \ldots, w_{k-1}, w_k]$
  $s_{k-1} \leftarrow$ confidence of segmenting before w_k
  $S \leftarrow S + [s_{k-1}]$ ▷ assume $S = [s_0, s_1, \ldots, s_{k-1}]$
  $B \leftarrow$ apply segmentation strategy to S ▷ assume $B = [b_0, b_1, \ldots, b_{k-1}]$
  if $b_i = 1$ $(0 \leq i \leq k-1)$ **then**
    output $[w_0, w_1, \ldots, w_i]$ as a segment
    remove first i elements from W and S
  end if
end for

segmentation strategy outputs no boundary, no action is taken (represented by (c) in Figure 2). Figure 2, a segment will be output and the inner sequence will be updated accordingly (as in the process represented by (d) in Figure 2).

The following two subsections describe the boundary confidence scores and segmentation strategies in detail, respectively.

3.1 Segment Boundary Confidence Score

This confidence score is based on an N-gram language model. Suppose the language model order is n.

The confidence score represents the plausibility of placing a sentence boundary after the word w_i, that is, converting the stream of words into $\cdots, w_{i-1}, w_i, \langle/s\rangle, \langle s\rangle, w_{i+1}, \cdots$, where $\langle/s\rangle$ and $\langle s\rangle$ are sentence start and end markers. The confidence score is based on the ratio of two probabilities arising from two hypotheses defined below:

Hypothesis I : there is no sentence boundary after word w_i. The corresponding Markov chain is,

$$\begin{aligned} P_i^{\langle \mathrm{I} \rangle} &= P_{\text{left}} \cdot P(w_{i+1}^{i+n-1}) \cdot P_{\text{right}} \\ &= P_{\text{left}} \cdot \prod_{k=i+1}^{i+n-1} p(w_k | w_{k-n+1}^{k-1}) \cdot P_{\text{right}} \end{aligned} \tag{1}$$

where p denotes the probability from the language model, P_{left} and P_{right} are the probabilities of the left and right contexts, of the words w_{i+1}^{i+n-1}.

Hypothesis II : there is a sentence boundary after the word w_i. The corresponding Markov chain is,

$$\begin{aligned} P_i^{\langle \mathrm{II} \rangle} &= P_{\text{left}} \cdot P(\langle/s\rangle, \langle s\rangle, w_{i+1}^{i+n-1}) \cdot P_{\text{right}} \\ &= P_{\text{left}} \cdot p(\langle/s\rangle | w_{i-n+2}^{i}) \cdot p(w_{i+1} | \langle s\rangle) \cdot \\ &\quad \prod_{k=i+2}^{i-n+1} p(w_k | w_{i+1}^{k-1}, \langle s\rangle) \cdot P_{\text{right}} \end{aligned} \tag{2}$$

The confidence score is defined as the ratio of the probabilities $P_i^{\langle \mathrm{II} \rangle}$ and $P_i^{\langle \mathrm{I} \rangle}$, that is,

$$\begin{aligned} s_i &= \frac{P_i^{\langle \mathrm{II} \rangle}}{P_i^{\langle \mathrm{I} \rangle}} \\ &= p(\langle/s\rangle | w_{i-n+2}^{i}) \cdot \frac{p(w_{i+1} | \langle s\rangle)}{p(w_{i+1} | w_{i-n-2}^{i})} \cdot \prod_{k=i+2}^{i-n+1} \frac{p(w_k | w_{i+1}^{k-1}, \langle s\rangle)}{p(w_k | w_{k-n+1}^{k-1})} \end{aligned} \tag{3}$$

This formula requires a context of $n-1$ future words w_{i+1}^{i+n-1}. This requirement causes a delay of $n-1$ words. If only one future word is used, this delay can be reduced to one word, formulated as,

$$s_i \approx p(\langle/s\rangle|w_{i-n+2}^{i}) \cdot \frac{p(w_{i+1}|\langle s\rangle)}{p(w_{i+1}|w_{i-n+2}^{i})} \tag{4}$$

Experimental results show that this approximation does not degrade the end-to-end translation quality (see Section 4.3). This might be because, for most languages, the next word w_{i+1} is the most informative to predict whether or not there is a sentence boundary after w_i.

3.2 Segmentation Strategies

In this subsection, two basic segmentation strategies that are based on a threshold heuristic and a latency heuristic, respectively, are first introduced. Then a hybrid strategy that combines these two heuristics is proposed in order to achieve lower delay.

3.2.1 Threshold-based Segmentation Strategy

The threshold-based strategy has a preset threshold parameter denoted: θ_{Th}. The strategy places sentence boundaries where the confidence score exceeds the threshold, formulated as,

$$b_i = \begin{cases} 1 & \text{if } s_i \geqslant \theta_{\text{Th}}, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

3.2.2 Latency-based Segmentation Strategy

The latency-based strategy has a maximum latency parameter denoted: θ_{ML}. Once the stream of confidence scores grows to a length of θ_{ML}, the strategy searches for the maximum confidence score in the stream of scores, and places a sentence boundary there, formulated as,

$$b_i = \begin{cases} 1 & \text{if } s_i \geqslant s_j (0 \leqslant j < \theta_{\text{ML}}), \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

3.2.3 Threshold-latency-based Segmentation Strategy

Both the threshold-based and latency-based segmenters have strengths and weakness with respect to time efficiency. The threshold-based strategy places a sentence boundary immediately when a confidence score exceeds the threshold. In this case, the delay is low. However, continuous sequences of low confidence scores, whose values are all under the threshold, will lead to a long unsegmented stream of words, resulting in high latency.

The latency-based strategy has the opposite behavior. The latency for words ranges from 0 to $\theta_{\text{ML}}-1$. The maximum latency is guaranteed to be $\theta_{\text{ML}}-1$, which is better than threshold-based strategy. But if there are some extremely high confidence scores in the stream, the latency-based strategy will ignore them, leading to unnecessarily long segments.

It is possible to combine to the threshold-based and latency-based strategies to achieve a lower delay. This hybrid threshold-latency-based strategies operates as follows:

- Apply the threshold heuristic to the stream of confidence scores. If a sentence boundary is predicted, then accept the boundary and update the stream.
- If the length of stream grows to θ_{ML}, apply the latency heuristic.

The method is formulated as,

$$b_i = \begin{cases} 1 & \text{if } s_i \geqslant \theta_{\text{Th}}, \\ 1 & \text{if } s_j < \theta_{\text{Th}} \text{ and } s_i \geqslant s_j (0 \leqslant j < \theta_{\text{ML}}), \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Corpus	# Sent. Pairs	Japanese		English	
		# Tokens†	# Words	# Tokens†	# Words
Training	5,134,941	106,044,671	93,672,553	84,371,311	74,733,865
Develop	6,000	150,690	141,036	103,473	95,054
Test	6,000	150,751	141,035	103,638	95,176

Table 1: Experimental Corpora.† Including punctuations.

4 Experiments

4.1 Experimental Settings

Experiments were performed on translation between Japanese and English in both directions. The word orders of these two languages are very different, thus long-distance reordering is often obligatory during translation. This makes simultaneous interpretation a very challenging task, and therefore we choose this language pair for experiments.

The experimental corpus was a union of corpora from multiple sources, including shared tasks such as the Basic Travel Expression Corpus (Takezawa et al., 2002), the NTCIR Patent Machine Translation Corpus (Goto et al., 2013), crawled web data and several in-house parallel resources. Table 1 shows the statistics of sentences and words in the training, development and test sets.

The corpora were pre-processed using standard procedures for MT. The Japanese text was segmented into words using Mecab (Kudo, 2005). The English text was tokenized with the tokenization script released with the Europarl corpus (Koehn, 2005) and converted to lowercase.

Two treatments were applied to the development and test sets in order to simulate the output from ASR engines. First, because ASR engines normally do not output punctuation, punctuation was removed. Second, because ASR engines output steams of tokens which are split by long pauses that may contain a few sentences, a random number (from 1 to 10) of sentences were concatenated to form the input.

After segmentation using the proposed methods, punctuation was inserted into the sentences with a hidden N-gram model model (Stolcke et al., 1998; Matusov et al., 2006) prior to translation. In (Anonymous, 2016), this method was shown to be the most effective strategy for the translation of unpunctuated text.

The time efficiency of segmenters were measured by average latency per source word using the definition given in (Finch et al., 2014). The quality of segmenters were measured by the BLEU of end-to-end translation, and because the segmented source sentences did not necessarily agree with the oracle, translations were aligned to reference sentences through edit distance in order to calculate BLEU (Matusov et al., 2005).

The parameters (all of the θ's in the 'Parameters' column in Table 2) were set by grid search to maximize the BLEU score on the development set. 5-gram interpolated modified Kneser-Ney smoothed language models were used to calculate the confidence. These were trained on the training corpus using the SRILM (Stolcke and others, 2002) tools. The machine translation system was an in-house phrase-based system that pre-ordered the input.

4.2 Experimental Results

The performance of the interpretation systems using different sentence segmenters is presented in Table 2. The following observations can be made.

First, the three proposed online sentence segmenters – the threshold-based, latency-based and hybrid ones – work reasonably well. They are much better than the trivial method of fixed-length segmentation, and comparable to the offline method using hidden N-gram models and also to the oracle sentence segmentation.

Second, the proposed threshold-latency-based segmenter consistently outperformed the threshold-based and latency-based segmenters in terms of both end-to-end translation quality and time efficiency.

Third, for Japanese-to-English translation, the threshold-based segmenter outperformed the latency-based segmenter. The reason might be that Japanese language has obvious end of sentence indicators such as "MA SU" and "DE SU", and the segmentation confidence scores immediately following them

Sentence Segmenter	Parameters	Dev. Set		Test Set	
		BLEU	Latency	BLEU	Latency
Japanese-to-English					
Oracle		13.82	NA	13.67	NA
Hidden N-gram †	θ_{Bias}=2.6	13.30	NA‡	12.97	NA‡
Fixed-length	θ_{Len}=36	11.71	16.66	11.55	16.63
Threshold-based	θ_{Th}=$e^{0.0}$	**13.38**	14.20	13.16	13.68
Latency-based	θ_{ML}=30	13.21	18.04	13.20	18.03
Threshold-latency	θ_{Th}=$e^{0.0}$, θ_{ML}=38	**13.38**	**12.98**	**13.28**	**12.89**
English-to-Japanese					
Oracle		13.84	NA	14.15	NA
Hidden N-gram†	θ_{Bias}=4.2	12.85	NA‡	13.10	NA‡
Fixed-length	θ_{Len}=18	11.86	8.19	12.15	8.20
Threshold-based	θ_{Th}=$e^{-0.6}$	12.93	**7.13**	13.19	**7.18**
Latency-based	θ_{ML}=20	**13.18**	12.25	13.38	12.26
Threshold-latency	θ_{Th}=$e^{0.2}$, θ_{ML}=20	**13.18**	10.01	**13.42**	10.11

Table 2: Performance of interpretation systems that use different sentence segmenters. The confidence scores in threshold-based, latency-based and threshold-latency-based segmenters were calculated using Equation 4. † Employed the segment tool from the SRILM toolkit. ‡ The method is not online since it operates on a whole sequence of words, thus the measurement of latency is not applicable.

# Future Words	Parameters	Dev. Set		Test Set	
		BLEU	Latency	BLEU	Latency
Japanese-to-English					
1	θ_{Th}=$e^{0.0}$, θ_{ML}=38	13.38	12.98	**13.28**	12.89
2	θ_{Th}=$e^{-0.2}$, θ_{ML}=38	**13.42**	**12.86**	13.21	**12.77**
3	θ_{Th}=$e^{-0.2}$, θ_{ML}=46	13.40	13.88	13.22	13.80
4	θ_{Th}=$e^{-0.2}$, θ_{ML}=46	13.38	14.71	13.23	14.65
English-to-Japanese					
1	θ_{Th}=$e^{0.2}$, θ_{ML}=20	**13.18**	10.01	13.42	10.11
2	θ_{Th}=$e^{0.2}$, θ_{ML}=18	13.12	**9.92**	**13.44**	**9.99**
3	θ_{Th}=$e^{0.4}$, θ_{ML}=22	13.14	12.78	13.41	12.78
4	θ_{Th}=$e^{0.4}$, θ_{ML}=22	13.17	13.65	13.41	13.65

Table 3: Performance of using different numbers of future words to calculate confidence scores.

will be quite high, allowing the threshold-based segmenter to easily identify the corresponding segment boundaries.

4.3 Confidence Scores Using Different Numbers of Future Words

Confidence scores were calculated using a context of up to four future words, as shown in Equation 3. The results are presented in Table 3. Though there is some randomness due to variance on the parameters chosen by grid search, these results show that using more future words does not effectively improve the quality of end-to-end translation, and tends to increase the latency, for the language pair of English and Japanese. Therefore, we found it sufficient to use just one future word.

4.4 Analysis

Table 4 presents an example of the proposed threshold-latency-based sentence segmenter in English-to-Japanese interpretation. The oracle segments of the input in this example are three sentences. The proposed method segments the input into four sentences, two of which are correct. The error is that the third oracle sentence is split into two sentences.

In this example, the proposed segmenter works reasonable accurately, as it recognized two sentences correctly out of three. Here, “i myself think”, “but it ’s ”, “we figured” are typical sentence beginnings in English, which can be recognized by language model. Therefore, the proposed language-model-based segmenters can correctly segment them. The error of splitting “we figured the ultimate test would be . . . ” into two sentences may have arisen from the fact that “we figured” occurs at the end of sentences, or

Input	i myself think that the argument about smoking is a slightly misleading one but it 's not predicted to go as high as once believed we figured the ultimate test would be to ask the dog 's owner to leave
Oracle Segments	<s> i myself think that the argument about smoking is a slightly misleading one </s> <s> but it 's not predicted to go as high as once believed </s> <s> we figured the ultimate test would be to ask the dog 's owner to leave </s>
Oracle Translation	<s> 私 自身 が 考える の は 喫煙 について の 議論 は 少し 誤解 を 招く もの だ という こと です </s> <s> しかし 予測 でき ない の は その 高 さ が かつて 思わ れ て い た の と 同じ 位 に なる という こと です </s> <s> 我々 が 考え た の は 最終 的 な テスト は 犬 の 所有 者 に 退去 する よう 依頼 する こと です </s>
Predicted Segments	<s> i myself think that the argument about smoking is a slightly misleading one </s> <s> but it 's not predicted to go as high as once believed </s> <s> we figured </s> <s> the ultimate test would be to ask the dog 's owner to leave </s>
Machine Translation	<s> 私 は 自分 の 喫煙 に関する 議論 が 少し 誤解 を 招く こと と 思い ます </s> <s> しかし 信じ られる よう に 高く なる こと が予測 さ れ て い ませ ん </s> <s> 私 たち が予想 し て い た </s> <s> たら 最終 的 な テスト は 犬 の 所有 者 に 出発 する の だ と 考え られ て い ませ ん </s>

Table 4: Example of Threshold-Latency-based Sentence Segmentor.

"the ultimate test would be" occurs as a sentence beginning in the training corpus of the language model. A language model can only capture local patterns, and cannot understand structures of compound sentences. This is a weakness of applying n-gram language modeling techniques to sentence segmentation. As a solution, it may be advantageous to replace the n-gram models with recurrent neural networks that are strong at exploiting long-distant context, and we plan to explore this in the future. It is interesting to note that, the resulting translations of the wrong segmentation "we figured" and "the ultimate test would be ..." are decent, as the origin meaning is delivered. This was an unexpected bonus that we owe to the evaluation framework. The evaluation framework in this paper is end-to-end BLEU, and places no constraints on segmentation positions. This helped to tune the parameters of the proposed methods properly. To sum up, this example illustrates that the proposed methods work reasonably well, and the evaluation framework itself is also making a contributions. However, errors caused by lack of understanding whole sentence structure are inevitable, and these need to be addressed in future work.

5 Conclusion

This paper proposed and studied a segmentation boundary confidence score and a set of online segmentation strategies for simultaneous interpretation. The solution expressed by Equations 4 and 7 was proven empirically to be both effective and efficient.

The choice to use sentence segmentation units was motivated by the desire to handle difficult language pairs that require long-distance intra-sentential word re-ordering (for example the Japanese-English pair studied in this paper). For these cases, using smaller units than sentences will prevent the translation system from being able to correctly re-order the words. For easier language pairs, segments shorter than sentences may be preferable; note that the proposed confidence score can be easily modified to handle sub-sentential segments if necessary. We would like to study this in the context of other language pairs in the future.

The primary motivation for this work was to create an online version of the hidden n-gram approach (Stolcke and Shriberg, 1996; Stolcke et al., 1998), a de facto standard method that is often used for sentence segmentation due to its effectiveness, simplicity and speed. However, it has a latency issue that prevents it from being used in simultaneous interpretation. The proposed method alleviates this latency issue while preserving all its merits, and we show empirically that the new method maintains the effectiveness of the hidden n-gram method even when the future context is reduced as far as a single word. We believe that the proposed method will not only lead to workable systems, but also establish

a meaningful baseline for related research. In the long term, we plan to incorporate the findings in this paper into an industrial simultaneous interpretation system.

References

Anonymous. 2016. A study of punctuation handling for speech-to-speech translation. In *22nd Annual Meeting on Natural Language Processing*, page to appear.

Srinivas Bangalore, Vivek Kumar Rangarajan Sridhar, Prakash Kolan, Ladan Golipour, and Aura Jimenez. 2012. Real-time incremental speech-to-speech translation of dialogs. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 437–445. Association for Computational Linguistics.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Andrew Finch, Xiaolin Wang, and Eiichiro Sumita. 2014. An Exploration of Segmentation Strategies in Stream Decoding. In *IWSLT*.

Christian Fügen, Alex Waibel, and Muntsin Kolss. 2007. Simultaneous translation of lectures and speeches. *Machine Translation*, 21(4):209–252.

Tomoki Fujita, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2013. Simple, lexicalized choice of translation timing for simultaneous speech translation. In *INTERSPEECH*, pages 3487–3491.

Isao Goto, Ka Po Chow, Bin Lu, Eiichiro Sumita, and Benjamin K Tsou. 2013. Overview of the patent machine translation task at the ntcir-10 workshop. In *Proceedings of the 10th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, NTCIR-10*.

Thanh-Le Ha, Jan Niehues, Eunah Cho, Mohammed Mediani, and Alex Waibel. 2015. The KIT Translation Systems for IWSLT 2015. In *Proceedings of the twelveth International Workshop on Spoken Language Translation (IWSLT), Da Nang, Veitnam*, pages 62–69.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*, volume 5, pages 79–86.

Taku Kudo. 2005. Mecab: Yet another part-of-speech and morphological analyzer. *http://mecab. sourceforge. net/*.

Michaeel Kzai, Brian Thompson, Elizabeth Salesky, Timothy Anderson, Grant Erdmann, Eric Hansen, Brian Ore, Katherine Young, Jeremy Gwinnup, Michael Hutt, and Christina May. 2015. The MITLL-AFRL IWSLT 2015 Systems. In *Proceedings of the twelveth International Workshop on Spoken Language Translation (IWSLT), Da Nang, Veitnam*, pages 23–30.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

Wei Lu and Hwee Tou Ng. 2010. Better punctuation prediction with dynamic conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 177–186. Association for Computational Linguistics.

Evgeny Matusov, Gregor Leusch, Oliver Bender, Hermann Ney, et al. 2005. Evaluating machine translation output with automatic sentence segmentation. In *IWSLT*, pages 138–144. Citeseer.

Evgeny Matusov, Arne Mauser, and Hermann Ney. 2006. Automatic sentence segmentation and punctuation prediction for spoken language translation. In *IWSLT*, pages 158–165.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2014. Optimizing segmentation strategies for simultaneous speech translation. In *ACL (2)*, pages 551–556.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Stephan Peitz, Markus Freitag, Arne Mauser, and Hermann Ney. 2011. Modeling punctuation prediction as machine translation. In *IWSLT*, pages 238–245.

Andreas Stolcke et al. 2002. Srilm-an extensible language modeling toolkit. In *INTERSPEECH*.

Andreas Stolcke and Elizabeth Shriberg. 1996. Automatic linguistic segmentation of conversational speech. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 2, pages 1005–1008. IEEE.

Andreas Stolcke, Elizabeth Shriberg, Rebecca A Bates, Mari Ostendorf, Dilek Hakkani, Madelaine Plauche, Gökhan Tür, and Yu Lu. 1998. Automatic detection of sentence boundaries and disfluencies based on recognized words. In *ICSLP*, pages 2247–2250.

Toshiyuki Takezawa, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto. 2002. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *LREC*, pages 147–152.

Lawrence Venuti. 2012. *The translation studies reader*. Routledge.

Mahsa Yarmohammadi, Vivek Kumar Rangarajan Sridhar, Srinivas Bangalore, and Baskaran Sankaran. 2013. Incremental segmentation and decoding strategies for simultaneous translation. In *IJCNLP*, pages 1032–1036.