Using Spoken Word Posterior Features in Neural Machine Translation

Kaho Osamura[1], *Takatomo Kano*[1], *Sakriani Sakti*[1,2], *Katsuhito Sudoh*[1,2], *Satoshi Nakamura*[1,2]

[1]Nara Institute of Science and Technology, Japan
[2]RIKEN, Center for Advanced Intelligence Project AIP, Japan
{osamura.kaho.oe5, kano.takatomo.km0, ssakti, sudoh, s-nakamura}@is.naist.jp

Abstract

A spoken language translation (ST) system consists of at least two modules: an automatic speech recognition (ASR) system and a machine translation (MT) system. In most cases, an MT is only trained and optimized using error-free text data. If the ASR makes errors, the translation accuracy will be greatly reduced. Existing studies have shown that training MT systems with ASR parameters or word lattices can improve the translation quality. However, such an extension requires a large change in standard MT systems, resulting in a complicated model that is hard to train. In this paper, a neural sequence-to-sequence ASR is used as feature processing that is trained to produce word posterior features given spoken utterances. The resulting probabilistic features are used to train a neural MT (NMT) with only a slight modification. Experimental results reveal that the proposed method improved up to 5.8 BLEU scores with synthesized speech or 4.3 BLEU scores with the natural speech in comparison with a conventional cascaded-based ST system that translates from the 1-$best$ ASR candidates.

1. Introduction

Spoken language translation is one innovative technology that allows people to communicate by speaking in their native languages. However, translating a spoken language, in other words, recognizing speech and then translating words into another language, is incredibly complex. A standard approach in speech-to-text translation systems requires effort to construct automatic speech recognition (ASR) and machine translation (MT), both of which are trained and tuned independently.

ASR systems, which aim for the perfect transcription of utterances, are trained and tuned by minimizing the word error rate (WER) [1]. MT outputs are optimized and automatically measured based on a wide variety of metrics. One of the standard methods is the BLEU metric. However, all the errors from the words in ASR outputs are treated uniformly without considering their syntactic roles, which are often critical for MT. Many studies have investigated the effectiveness of the WER metric of ASR on the whole speech translation pipeline [2, 3, 4] and verified that ASR errors that compose the WER metric do not contribute equally to the BLEU score of translation quality.

Furthermore, most MT systems are only trained and optimized using error-free text data. Despite the fact that ASR technologies and their recognition rates have continued to improve, the occurrence of speech recognition errors remains inevitable. This is because there are many ambiguities due to a wide variety of acoustic and linguistic patterns produced by different speakers with various speaking styles and background noises. If the ASR engine makes mistakes, the translation accuracy will be significantly reduced. Thus, ignoring the existence of ASR errors while constructing a speech translation system is practically impossible.

Previous research on traditional phrase-based MTs has attempted to train the ASR and MT parameters of the log-linear model to directly optimize the BLEU score of the translation metric of full speech translation systems [3]. It allows the model to directly select recognition candidates that are easy to translate and improve the translation accuracy given an imperfect speech recognition. Ohgushi et al. [5] further elaborated various techniques in the context of the joint optimization of ASR and MT, including minimum error rate training (MERT) [6], pair-wise ranking optimization (PRO) [7], and the batch margin infused relaxed algorithm (MIRA) [8]. Other studies directly performed translation on the lattice representations of the ASR output [9, 10, 11]. The results showed that a better translation can be achieved by translating the lattices rather than with the standard cascade system that translated the single best ASR output.

Recently, deep learning has shown great promise in many tasks. A sequence-to-sequence attention-based neural network is one type of architecture that offers a powerful model for machine translation and speech recognition [12, 13]. Several studies revisited similar problems and proposed handling lattice inputs by replacing the encoder part with a lattice encoder to obtain a lattice-to-sequence model [14, 15]. With these methods, robust translation to speech recognition errors became possible. However, this approach requires a large modification to standard NMT systems, resulting in a complicated model that is hard to train. Also, as the NMT takes word lattices as input, it might be difficult to backpropagate a translation error to the ASR part.

An extreme case is to train the encoder-decoder architecture for end-to-end speech translation (ST) tasks, which directly translates speech in one language into text in another. Duong et al. [16] directly trained attentional models on parallel speech data. But their work focused only on alignment performance. The works by Berard et al. [17] might be the first attempts that successfully build a full-fledged end-to-end attentional-based speech-to-text translation system. But they only performed with a small parallel French-English BTEC corpus, and their best results were behind the cascade baseline model. Later on, Weiss et al. [18] proposed a similar approach and conducted experiments on the Spanish Fisher and Callhome corpora of telephone conversations augmented with English translations. However, most of these works were only done for language pairs with similar syntax and word order (SVO-SVO), such as Spanish-English or French-English. For such languages, only local movements are sufficient for translation. Kano et al. [19] showed that direct attentional ST approach failed to handle English-Japanese language pairs with SVO versus SOV word order.

In this research, we also focus on English-Japanese and we aim for a neural speech translation that is robust against speech recognition errors without requiring significant changes in the NMT structure. This can be considered as a simplified version of the one that directly performed translation on the lattice representations. But, instead of providing full lattice outputs, we perform a neural sequence-to-sequence ASR as feature processing that is trained to produce word posterior features given spoken utterances. This might resemble the word confusion networks (WCNs) [20] that can directly express the ambiguity of the word hypotheses at each time point. The resulting probabilistic features are used to train NMT with just a slight modification. Such vectors are expected to express the ambiguity of speech recognition output candidates better than the standard way using the 1-$best$ ASR outputs while also providing a simpler structure than the lattice outputs. During training, the approach also allows backpropagating the errors from NMT to ASR and performs join training. Here, we evaluate our proposed English-Japanese speech translation model using both synthesized and natural speech with various degrees of ASR errors.

2. Overview of Attention-based Speech Translation

Our English-Japanese end-to-end speech translation system consists of ASR and MT modules that were constructed on standard attention-based, encoder-decoder neural network architecture [21, 22].

2.1. Basic Attentional Encoder-Decoder model

An attentional encoder-decoder model consists of an encoder, a decoder, and attention modules. Given input sequence $\boldsymbol{x} = [x_1, x_2, ..., x_N]$ with length N, the encoder produces a sequence of vector representation $h^{enc} = (h_1^{enc}, h_2^{enc}, ..., h_N^{enc})$. Here, we used a bidirectional recurrent neural network with long short-term memory (bi-LSTM) units [23], which consist of forward and backward LSTMs. The forward LSTM reads the input sequence from x_1 to x_N and estimates forward $\overrightarrow{h^{enc}}$, and the backward LSTM reads the input sequence in reverse order from x_N to x_1 and estimates backward $\overleftarrow{h^{enc}}$. Thus, for each input x_n, we obtain h_n^{enc} by summation forward $\overrightarrow{h^{enc}}$ and backward $\overleftarrow{h^{enc}}$:

$$h_n^{enc} = \overrightarrow{h_n^{enc}} + \overleftarrow{h_n^{enc}}. \tag{1}$$

The decoder, on the other hand, predicts target sequence $\boldsymbol{y} = [y_0, y_1, y_2, ..., y_M]$ with length M by estimating conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$. Here, we use uni-directional LSTM (forward only). Conditional probability $p(\boldsymbol{y}|\boldsymbol{x})$ is estimated based on the whole sequence of the previous output:

$$p(y_m|\boldsymbol{y}_{<m}, \boldsymbol{x}) = \text{softmax}(W_y \tilde{h}_m^{dec}). \tag{2}$$

Decoder hidden activation vector $\tilde{h}_m^{dec}$ is computed by applying linear layer W_c over context information c_m and current hidden state h_m^{dec}:

$$\tilde{h}_m^{dec} = \tanh(W_c[c_m; h_m^{dec}]). \tag{3}$$

Here, c_m is in the context information of the input sequence when generating current output at time m. It is estimated by the attention module over encoder hidden states h_n^{enc}:

$$c_m = \sum_{n=1}^{N} a_m(n) * h_n^{enc}, \tag{4}$$

where variable-length alignment vector a_m is computed whose size equals length of input sequence x:

$$\begin{aligned} a_m &= \text{align}(h_n^{enc}, h_m^{dec}) \\ &= \text{softmax}(\text{dot}(h_n^{enc}, h_m^{dec})). \end{aligned} \tag{5}$$

This step assists the decoder to find relevant information on the encoder side based on the current decoder hidden states. Several variations calculate align(h_n^{enc}, h_m^{dec}). Here, we simply use the dot product between the encoder and decoder hidden states.

2.2. Automatic Speech Recognition

Speech recognition tasks estimate a word sequence given a sequence of speech features. Input sequence $\boldsymbol{x} = [x_1, ..., x_N]$ is the input speech filter bank feature sequence of the source language, and target sequence $\boldsymbol{y} = [y_1, ..., y_M]$ is the predicted corresponding word sequence in the source language.

2.3. Machine Translation

Machine translation tasks estimate a word sequence of a target language given a word sequence of a source language. Input sequence $\boldsymbol{x} = [x_1, ..., x_N]$ is the word sequence of the source language, and target sequence $\boldsymbol{y} = [y_1, ..., y_M]$ is the predicted corresponding word sequence in the target language. Here, x_n is a one-hot vector in the baseline or posterior vector in the proposed method, y_m is the index representation of the words, and y_0 is an index representation of the target sequence's start.

2.4. Speech-to-text Translation

Speech-to-text translation tasks estimate a word sequence of a target language given a sequence of speech features. Here, we use both the sequence-to-sequence ASR and MT systems. Output sequence $\boldsymbol{y}$ from ASR becomes input sequence $\boldsymbol{x}$ in an MT system.

3. Proposed method: NMT using Spoken Word Posterior Features

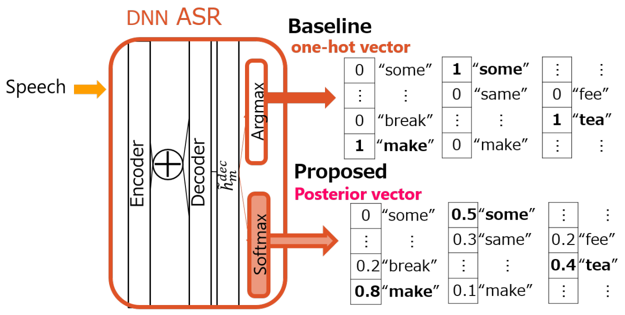

Figure 1: Construction of spoken word posterior features

Fig. 1 illustrates the construction of spoken word posterior features. Here, we train an end-to-end ASR using the standard attention-based encoder-decoder neural network architecture described in the previous section. But instead of providing 1-*best* outputs of the most probable word sequence to the translation system,

$$\hat{y}_m = \underset{y_m}{\operatorname{argmax}}\ p(y_m|\boldsymbol{y}_{<m}, \boldsymbol{x}), \tag{6}$$

we utilize the posterior probability vectors before the argmax function:

$$p(y_m|\boldsymbol{y}_{<m}, \boldsymbol{x}). \tag{7}$$

This way the vectors can still express the ambiguity of the speech recognition output candidates with probabilities.

The resulting probabilistic features are then used to train the NMT with only a slight modification. We train the end-to-end NMT using the standard attention-based encoder-decoder neural network architecture described in the previous section. The only difference is in the input features. Instead of training the model with the one-hot vector of the most probable words, we utilize the posterior vectors obtained from the ASR. However, the dimension of input vector representation used in a standard one-hot vector and the proposed posterior vectors is the same. The overall architecture is illustrated in Fig. 2.

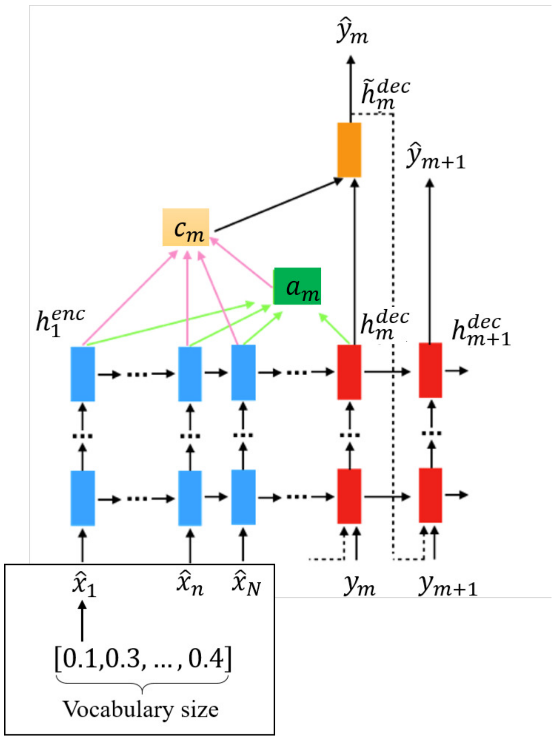

Figure 2: Proposed NMT architecture

4. Experiments

We evaluated the performance of the proposed method on an English-Japanese translation task. To simulate the effect of various ASR errors, we first assessed it on synthesized speech and later applied it to natural speech.

4.1. Data set

The experiments were conducted using a basic travel expression corpus (BTEC) [24]:

- **Text corpus**
  We used a BTEC English-Japanese parallel text corpus that consists of about 460k (BTEC1-4) training sentences and 500 sentences in the test set.
- **Synthesized speech corpus**
  Since corresponding speech utterances for the BTEC parallel text corpus are not available, we used Google text-to-speech synthesis [25] to generate a speech corpus of the BTEC1 source language (about 160k utterances). We used about 500 speech utterances in the test set.
- **Natural speech corpus**
  We also evaluated with natural speech. In this case, we used the ATR English speech corpus [26] in our experiments. The text material was based on the basic travel expression domain. The speech corpus we used consisted of American, British, and Australian (AUS)

English accents with about 120k utterances spoken by 100 speakers (50 males, 50 females) for each accent.

The speech utterances were segmented into multiple frames with a 25-ms window size and a 10-ms step size. Then we extracted 23-dimension filter bank features using Kaldi's feature extractor [27] and normalized them to have zero mean and unit variance.

4.2. Models

We further used the data to build a speech translation system with attention-based ASR and MT systems. The ASR and NMT share the same vocabulary (16,745 words). The dimensions of the distributed vector representation are smaller than vocabulary size (the size depends on the model settings). The hidden encoder and decoder layer consists of 500 nodes. A batch size of 32 and a dropout of 0.1 were also applied. For all systems, we used a learning rate of 0.0001 for the encoder and 0.0005 for the decoder and adopted Adam [28] to all the models.

As we aim to have a neural speech translation that is robust against speech recognition errors without requiring significant changes in the NMT structure. We constructed three types of models that fit those requirements:

- **Text-based machine translation system (upper-bound)**
  This is a text-to-text translation model from the source language to the target language. Here the BTEC English-Japanese parallel text corpus is used to train the model.

- **Baseline speech translation**
  This speech-to-text translation model was created by cascading the ASR (speech-to-text) in the source language with a text-to-text MT module using 1-*best* ASR outputs. First, we pre-trained the NMT with the BTEC English-Japanese parallel text corpus and then fine-tuned the NMT model with a one-hot vector provided from the ASR.

- **Proposed speech translation**
  This speech-to-text translation model was created by cascading ASR (speech-to-text) in the source language with the text-to-text MT module using the ASR posterior vectors. First, we pre-trained the ASR with the speech of the source language and the NMT with the BTEC English-Japanese parallel text corpus. After that, we fine-tuned the parameter of both models by jointly training, where the posterior vector of ASR output is used as the NMT input.

Note that the ASR systems used for the baseline and the proposed systems are the same. Also, all translation systems were tuned adequately, and the best model from training epochs was selected for each system.

5. Result

5.1. Speech Recognition System

To simulate different degrees of ASR errors, we constructed an ASR model using synthesized speech with different numbers of training epochs, resulting in four different models with the following WERs: (1) System 1 (WER=15.17%), System 2 (WER=12.34%), System 3 (WER=11.05%), and System 4 (WER=8.82%). As a model that is trained with natural speech, our performance achieved a 24.98% WER.

5.2. Translation System

As mentioned earlier, we compare three translation system: one for standard text-based machine translation, one for baseline speech translation with the cascade model, and one for our proposed speech translation.

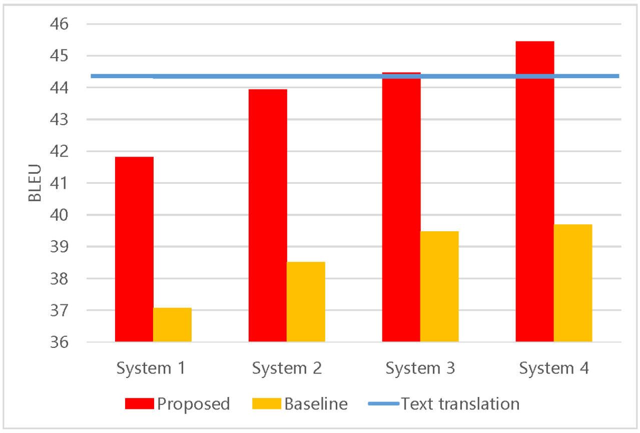

Figure 3: Translation quality given synthesized speech input

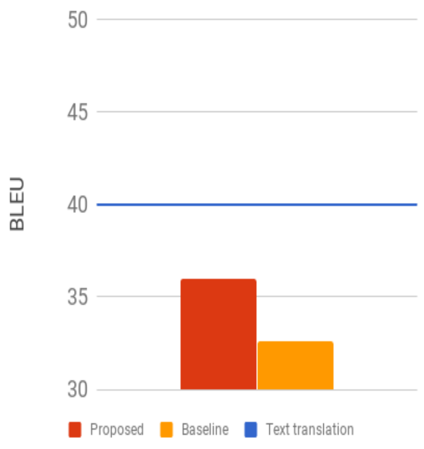

Figure 4: Translation quality given natural speech input

The quality of those translation systems with the input of synthesized speech was evaluated using BLEU [29] and

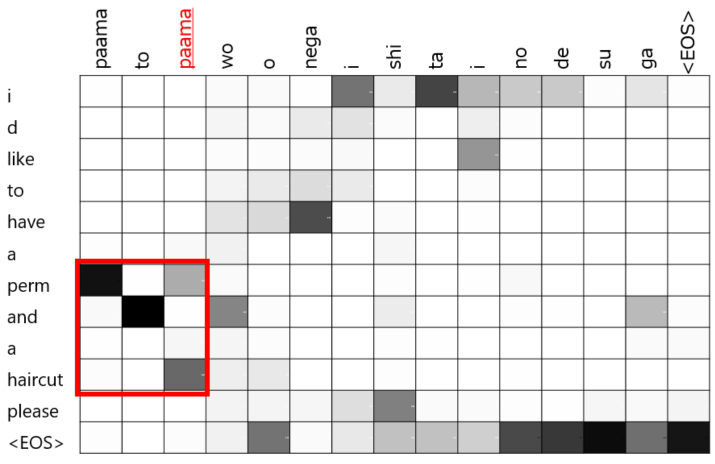

Figure 5: Attention matrix of text translation

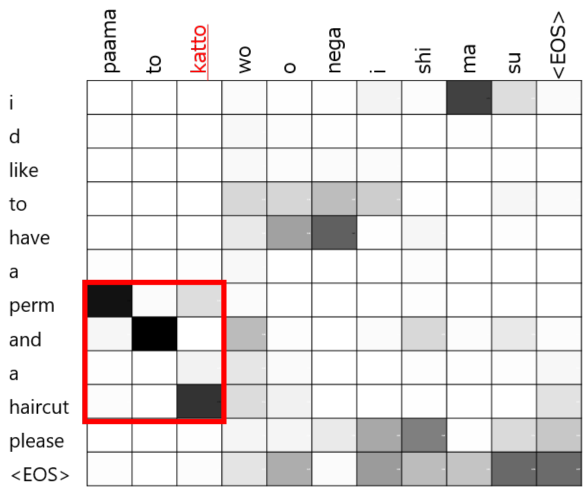

Figure 6: Attention matrix of proposed method

shown in Fig. 3. Here System 1-4 represent of using different ASR systems (1-4), respectively. The results show that the better the ASR performance, the stronger the baseline cascade model. Nevertheless, our proposed approach stable outperformed the cascade model in all cases. The BLEU score improved from 4.8 to 5.8 compared to the baseline model.

The proposed methods (System 3 and 4) exceed the text translation because the recognition candidates included in the posterior vector made it possible to correctly distinguish confusing words in the word embedding of the text translation. We will scrutinize this result in the next section.

Next, the quality of the speech translation systems using natural speech was also evaluated using BLEU and shown in Fig. 4. For the text translation, we provided the transcription of the natural speech, which is different than the text used in Fig. 3. This system used the ASR model where WER is 24.98%. Importantly, unlike several published ASR systems using BTEC dataset, our ASR system only used the text transcription of the training set for the language model. Therefore, the ASR results reported in the paper could not reach state-of-the-art ASR performance. Nevertheless, the translation results are still convincing as evidence of the proposed framework's effectiveness. The proposed method improved the 4.3 BLEU score of the baseline model, confirming that the proposed method is also effective for natural speech.

6. Discussion

Table 1 shows the sentence output examples in English-Japanese translation: (1) with ASR error, and (2) without ASR error. In the first example, to analyze the effect of ASR error, we compare the sentence output of the proposed model and the baseline (the cascade model). Here, ASR misrecognized "shoe" as "station". This error impacted the baseline (cascade system), where it translated "station" as "`eki`" (the correct translation for "shoe store" is "`kutsuya`"). However, in the proposed method, it was still able to translate it to "`kutsuya`". This might be because the ASR provided a posterior vector in which the recognition candidate and each a posteriori probability are weighted (Table 2). Here, "shoe" information was still contained in the posterior vector with only slightly lower probability than "station," and based on the context information, the machine translation translated the word as "`kutsuya`."

In the second example, ASR provided a correct sentence. Here, we compare the sentence output of the proposed model and the text translation. Since the contexts of "perm" and "haircut" are close, the text translation mistakenly translated

Table 1: Examples of sentences output: (1) with ASR error, and (2) without ASR error.

Example 1: With ASR error	
ASR reference	Excuse me where is the closest shoe store?
ASR result	Excuse me where is the closest station store?
Baseline	`Sumimasen ichiban chikai eki wa doko desuka?`
Proposed	`Sumimasen ichiban chikai kutsuya wa doko desuka?`
MT reference	`Sumimasen ichiban chikai kutsuya wa doko desuka?`
Example 2: Without ASR error	
ASR reference	i d like to have a perm and a haircut please
ASR result	i d like to have a perm and a haircut please
Text translation	`Paama to paama o onegai shitai nodesuga`
Proposed	`Paama to katto o onegaishimasu`
MT reference	`Paama to katto o onegaishimasu`

Table 2: Posterior vector

Recognized	Posterior
station	0.439
shoe	0.321
change	0.086
cashier	0.036
always	0.016

both "perm" and "haircut" into "`paama`" (Fig. 5 illustrates the text translation's alignment matrix). On the other hand, having a posterior vector as the input in the proposed model (see the attention matrix in Fig. 6) allowed NMT to correctly distinguish confusing words by the word embedding of the text translation.

7. Conclusions

In this research, a speech translation system that is robust against speech recognition errors is obtained by using a posterior vector, which is a normalized vector that expresses the ambiguity of the speech recognition candidates, as the input of an NMT engine. The lower the WER of the ASR model is, the weaker the tendency of translation error becomes. Nevertheless, the whole test's accuracy surpassed the baseline. As a result, the posterior vector improved the BLEU score by 4.8 to 5.8 points over the baseline in the simulation experiment and improved it by 4.3 BLEU points over the baseline in the experiment using natural voice. By providing the probability of the speech recognition output candidates in speech translation, an optimal input selection for NMT was made. In the future, we will directly perform join training from ASR to NMT.

8. Acknowledgements

Part of this work was supported by JSPS KAKENHI Grant Numbers JP17H06101 and JP17K00237. We also thank Phillip Arthur, Do Quoc Truong and Andros Tjandra for their feedback and insightful discussions.

9. References

[1] X. He, L. Deng, and W. Chou, "Discriminative learning in sequential pattern recognition," *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 14–36, 2008.

[2] P. Dixon, A. Finch, C. Hori, and H. Kashioka, "Investigation on the effects of ASR tuning on speech translation performance," in *IWSLT*, San Francisco, USA, 2011.

[3] X. He, L. Deng, and A. Acero, "Why word error rate is not a good metric for speech recognizer training for the speech translation task?" in *ICASSP*, Prague, Czech Republic, 2011.

[4] N. Ruiz and M. Federico, "Assessing the impact of speech recognition errors on machine translation quality," in *Association for Machine Translation in the Americas (AMTA)*, 2014, pp. 261–274.

[5] M. Ohgushi, G. Neubig, S. Sakti, T. Toda, and S. Nakamura, "An empirical comparison of joint optimization techniques for speech translation," in *INTERSPEECH*, 2013, pp. 2619–2623.

[6] F. J. Och, "Minimum error rate training in statistical machine translation," in *ACL*, Sapporo, Japan, 2003.

[7] M. Hopkins and J. May, "Tuning as ranking," in *EMNLP*, Edinburgh, UK, 2011, pp. 1352–1362.

[8] C. Cherry and G. Foster, "Batch tuning strategies for statistical machine translation," in *NAACL*, Montreal, Canada, 2012, pp. 34–35.

[9] S. Saleem, S.-C. Jou, S. Vogel, and T. Schultz, "Using word lattice information for a tighter coupling in speech translation systems," in *ICSLP*, Jeju Island, Korea, 2004, pp. 41–44.

[10] R. Zhang, G. Kikui, H. Yamamoto, and W.-K. Lo, "A decoding algorithm for word lattice translation in speech translation," in *IWSLT*, Pittsburgh, USA, 2005, pp. 23–29.

[11] E. Matusov, B. Hoffmeister, and H. Ney, "ASR word lattice translation with exhaustive reordering is possible," in *Interspeech*, Brisbane, Australia, 2008, pp. 2342–2345.

[12] j. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *arXiv preprint arXiv:1506.07503*, 2015.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[14] J. Su, Z. Tan, D. Xiong, R. Ji, X. Shi, and Y. Liu, "Lattice-based recurrent neural network encoders for neural machine translation," in *AAAI*, 2017, pp. 3302–3308.

[15] M. Sperber, G. Neubig, J. Niehues, and A. Waibel, "Neural lattice-to-sequence models for uncertain inputs," in *EMNLP*, Copenhagen, Denmark, 2017, p. 13801389.

[16] L. Duong, A. Anastasopoulos, D. Chiang, S. Bird, and T. Cohn, "An attentional model for speech translation without transcription," in *HLT-NAACL*, 2016.

[17] A. Berard and O. Pietquin, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," in *30th Conference on Neural Information Processing Systems*, 2016.

[18] R. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," *arXiv preprint arXiv:1703.08581*, 2017.

[19] T. Kano, S. Sakti, and S. Nakamura, "Structured-based curriculum learning for end-to-end English-Japanese speech translation," in *INTERSPEECH*, 2017.

[20] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[21] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[22] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[25] "Google Text to Speech API," https://github.com/pndurette/gTTS.

[26] S. Sakti, M. Paul, A. Finch, X. Hu, J. Ni, N. Kimura, S. Matsuda, C. Hori, Y. Ashikari, H. Kawai, H. Kashioka, E. Sumita, and S. Nakamura, "Distributed speech translation technologies for multiparty multilingual communication," *TSLP*, vol. 9, pp. 4:1–4:27, 2012.

[27] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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