INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Transcribing Speech as Spoken and Written Dual Text Using an Autoregressive Model

Mana Ihori, Hiroshi Sato, Tomohiro Tanaka, Ryo Masumura, Saki Mizuno, Nabukatsu Hojo

NTT Computer and Data Science Laboratories, NTT Corporation, Japan
mana.ihori@ntt.com

Abstract

This paper proposes a novel method to jointly generate spoken and written text from input speech for expanding use cases of speech-based applications. The spoken text generated using speech-to-spoken text systems, i.e., speech recognition systems, has disfluencies and no punctuation marks. Thus, spoken text is often converted into written text using a spoken text-to-written text system. However, this cascading is unsuitable for overall optimization and computationally expensive. Although a speech-to-written-text system that directly outputs the written text from the speech is also developed, it cannot output the spoken text. To efficiently produce both spoken and written text from speech, our key advance is to handle a joint text of spoken and written texts in an autoregressive model. This enables us to correctly generate both spoken and written texts by considering their dependencies via a single decoding process. Our experiments demonstrate the effectiveness of the proposed method.

Index Terms: automatic speech recognition, spoken text-to-written text conversion, speech-to-written text

1. Introduction

Recent improvements to the performance of automatic speech recognition (ASR) using deep neural networks [1–3] have led to an increase in the number of applications that utilize ASR. In the applications, natural language processing technologies such as summarization [4, 5], machine translation [6, 7], and slot filling [8, 9] are utilized for subsequent processing. Spoken text that ASR models generate has disfluencies such as fillers and redundant expressions and no punctuation marks. Thus, spoken text is often converted into written text which is optimal for these natural language processing technologies. On the other hand, disfluencies also reflect speaker interaction [10], and it is important to analyze spoken text in applications such as interactive robots. In other words, it is necessary to output both spoken and written text. In this paper, we propose an efficient modeling method to output both spoken and written text with a focus on computation complexity and performance.

Conventionally, ASR results have been converted into written text using a spoken text-to-written text conversion (ST2WT) model [11, 12]. The ST2WT model is trained on multiple tasks, such as disfluency deletion and punctuation restoration, using paired spoken and written text data. In the ST2WT, the performance is degraded due to ASR errors because written text is not directly optimized from speech features. Also, although we can obtain spoken and written text from the ASR and ST2WT models, it requires double the number of model parameters. To output written text robustly to ASR errors, a speech-to-written text (S2WT) model that outputs written text from speech features has been proposed [13, 14]. However, the S2WT is a model for outputting written text, so spoken text cannot be obtained unless ASR is also separately performed.

To efficiently produce both the spoken and the written text from speech features, we investigate a method to output both texts in a single model. Our key idea is to generate written text after spoken text via a single decoding. This approach has two advantages. First, we can reduce the computational cost by outputting both the spoken and the written text with one model. Second, written text generation performance is improved by continuously outputting spoken and written text. We suppose that written text can be output correctly by using information from the spoken text as well as speech features because the written text can be generated from the spoken text as done in ST2WT. Therefore, we expect this approach to output both the spoken and the written text correctly with a lightweight model.

In this paper, we propose *speech-to-spoken and written text (S2SWT)*, a method to generate spoken and written dual text using a separator token [sep] (“spoken text [sep] written text”) from speech features in an autoregressive model. To use the common knowledge of ASR and S2WT, the S2SWT model is trained in a two-stage manner. In the first stage, the model is trained from ASR, S2WT, and S2SWT tasks jointly by distinguishing each task with special tokens in a single model. In the second stage, the model is fine-tuned using only the S2SWT task. In the two-stage training, the performance of ASR can be maintained because the model can also train to generate spoken text using a dataset without written text. Therefore, we expect the proposed method to be able to output both the spoken and the written text correctly via a single decoding process. To evaluate the proposed method, we conduct evaluation experiments on Japanese ASR and S2WT tasks.

Our main contributions are as follows:

- We propose a speech-to-spoken and written text that outputs spoken and written dual text from speech features in an autoregressive model. The S2SWT model is trained in a two-stage manner to use paired speech and spoken or written text for training.
- The proposed method provides efficient modeling for model parameter size and performance because it can simultaneously generate spoken and written text in a single model while considering the relationship between these texts.
- Our experiments on Japanese ASR and S2WT tasks using the Corpus of Spontaneous Japanese (CSJ) [15] demonstrate that the proposed method outperforms conventional methods.

2. Related work

Spoken text-to-written text conversion In the spoken-to-written conversion task, multiple tasks, such as disfluency dele-

10.21437/Interspeech.2023-1655

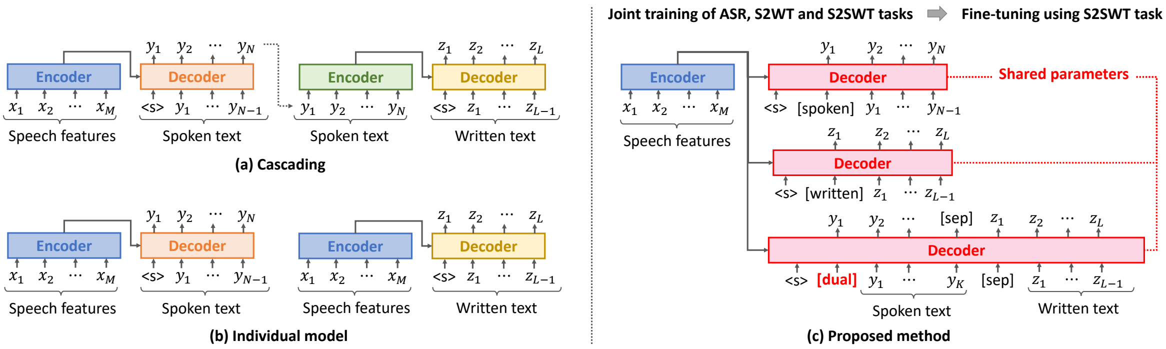

Figure 1: *Modeling methods to generate spoken and written text. (a) and (b) show the conventional methods, and (c) shows the proposed method.*

tion [16–18], capitalization and punctuation restoration [19–21], inverse text normalization [22, 23], ASR spelling correction [24], and grammar correction [25], have been performed using an individual text-to-text model. On the other hand, methods to handle these multiple tasks simultaneously have also been proposed to improve the readability of written text [11,12]. In this paper, the written text is generated using these multiple tasks simultaneously in the ST2WT and S2WT tasks.

Speech-to-written text Recent studies have examined ways to output written text from speech features directly. Nozaki at el. [13] proposed a method to perform a punctuation restoration task. Although this method aims to improve the performance of the punctuation restoration by using an auxiliary ASR task, no spoken text is output. Futami at el. [14] came up with a method that can output ASR and labels of disfluency detection simultaneously, but the parameter size is significantly increased because a decoder is required for each task. Also, the method is specialized for the disfluency deletion task, making it unsuitable for other S2WT tasks.

Joint model considering relationship between tasks In speech translation (ST), there is a method that performs ASR and ST using only speech features while considering the relationship between these tasks [26, 27]. ST is a task that converts the results of the ASR into translated text and is similar to ST2WT. In this method, two decoders for ASR and ST apply an attention mechanism for each other so that each task can be learned and inferred while using the information of the other. However, since this modeling requires an additional attention layer to refer to the information of each other, the number of parameters necessarily increases. Also, since ST and ASR are decoded in parallel, ST cannot take advantage of all the results of ASR, even though ST is the task that translates the results of ASR. In the proposed method, a model learns to output spoken and written text while considering the relationship between these texts but does not change the model architecture. Also, written text is generated using not some but all results of ASR.

3. Preliminaries

This section describes the cascading and individual models method to generate both the spoken and the written text. (a) and (b) in Figure 1 show the overview of these methods. In this paper, we define speech features as $\boldsymbol{X} = \{x_1, \cdots, x_m, \cdots, x_M\}$, where x_m is the m-th frame and M is the frame length. Also, we define spoken text as $\boldsymbol{Y} = \{y_1, \cdots, y_n, \cdots, y_N\}$ and written text as $\boldsymbol{Z} = \{z_1, \cdots, z_l, \cdots, z_L\}$, where y_n and z_l are the n-th and l-th tokens, and N and L is the number of tokens in the spoken and the written text, respectively.

3.1. Cascading

In the cascading, the ASR model output $\boldsymbol{Y}$ is used as input to the ST2WT model to generate $\boldsymbol{Z}$. The ASR model generates $\boldsymbol{Y}$ from $\boldsymbol{X}$, and the ST2WT model generates $\boldsymbol{Z}$ from $\boldsymbol{Y}$ as

$$P(\boldsymbol{Y}|\boldsymbol{X};\boldsymbol{\Theta}_{\rm asr}) = \prod_{n=1}^{N} P(y_n|y_1,\cdots,y_{n-1},\boldsymbol{X};\boldsymbol{\Theta}_{\rm asr}), \tag{1}$$

$$P(\boldsymbol{Z}|\boldsymbol{Y};\boldsymbol{\Theta}_{\rm st2wt}) = \prod_{l=1}^{L} P(z_l|z_1,\cdots,z_{l-1},\boldsymbol{Y};\boldsymbol{\Theta}_{\rm st2wt}), \tag{2}$$

where $\boldsymbol{\Theta}_{\rm asr}$ and $\boldsymbol{\Theta}_{\rm st2wt}$ are trainable model parameter sets in the ASR and ST2WT models, respectively.

Training: The loss function for ASR or ST2WT model is defined as

$$\mathcal{L}_{\rm asr} = -\sum_{(\boldsymbol{X},\boldsymbol{Y})\in\mathcal{D}_{\rm asr}} \log P(\boldsymbol{Y}|\boldsymbol{X};\boldsymbol{\Theta}_{\rm asr}), \tag{3}$$

$$\mathcal{L}_{\rm st2wt} = -\sum_{(\boldsymbol{Y},\boldsymbol{Z})\in\mathcal{D}_{\rm st2wt}} \log P(\boldsymbol{Z}|\boldsymbol{Y};\boldsymbol{\Theta}_{\rm st2wt}), \tag{4}$$

where $\mathcal{D}_{\rm asr}$ is a dataset that has the paired speech features and spoken text data, and $\mathcal{D}_{\rm st2wt}$ is a dataset that has the paired spoken and written text data.

Decoding: The decoding problem for spoken or written text is defined as

$$\hat{\boldsymbol{Y}} = \arg\max_{\boldsymbol{Y}} P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\Theta}_{\rm asr}), \tag{5}$$

$$\hat{\boldsymbol{Z}} = \arg\max_{\boldsymbol{Z}} P(\boldsymbol{Z}|\hat{\boldsymbol{Y}},\boldsymbol{\Theta}_{\rm st2wt}). \tag{6}$$

3.2. Individual model

The individual model generates $\boldsymbol{Y}$ or $\boldsymbol{Z}$ from $\boldsymbol{X}$ using ASR or S2WT model individually. $\boldsymbol{Y}$ is generated using the ASR model with Eq. (1). Also, $\boldsymbol{Z}$ is generated using the S2WT model as

$$P(\boldsymbol{Z}|\boldsymbol{X};\boldsymbol{\Theta}_{\mathrm{s2wt}}) = \prod_{l=1}^{L} P(z_l|z_1,\cdots,z_{l-1},\boldsymbol{X};\boldsymbol{\Theta}_{\mathrm{s2wt}}), \quad (7)$$

where $\boldsymbol{\Theta}_{\mathrm{s2wt}}$ is trainable model parameter set.

Training: The loss function for the S2WT model is defined as

$$\mathcal{L}_{\mathrm{s2wt}} = -\sum_{(\boldsymbol{X},\boldsymbol{Z})\in\mathcal{D}_{\mathrm{s2wt}}} \log P(\boldsymbol{Z}|\boldsymbol{X};\boldsymbol{\Theta}_{\mathrm{s2wt}}), \quad (8)$$

where $\mathcal{D}_{\mathrm{s2wt}}$ is the paired speech features and written text data. Also, the loss function for the ASR model is defined as Eq. (3).

Decoding: The decoding problems for spoken and written text are defined as

$$\hat{\boldsymbol{Y}} = \arg\max_{\boldsymbol{Y}} P(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\Theta}_{\mathrm{asr}}), \quad (9)$$

$$\hat{\boldsymbol{Z}} = \arg\max_{\boldsymbol{Z}} P(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\Theta}_{\mathrm{s2wt}}). \quad (10)$$

4. Proposed method

4.1. Speech-to-spoken and written text

We propose a speech-to-spoken and written text (S2SWT) to output spoken and written dual text from speech features in an autoregressive model. In the proposed method, spoken text is generated from speech features, and written text is generated from not only speech features but also spoken text. Our idea to generate written text from speech features and spoken text is to generate a joint text of spoken and written text autoregressively. The written text can be generated using the information of the spoken text by outputting it after the spoken text in an autoregressive model. Thus, we expect the S2SWT model to generate written text from speech features while considering the dependency on converting spoken text into written text. (c) in Figure 1 shows the overview of the proposed method.

4.2. Modeling method

In the proposed method, to make effective use of each spoken and written text data, we train ASR, S2WT, and S2SWT in a single model by distinguishing each task with special tokens. In the S2SWT task, the spoken and written dual text $\boldsymbol{W} = \{\boldsymbol{Y}, [\mathrm{sep}], \boldsymbol{Z}\} = \{y_1,\cdots,y_N,[\mathrm{sep}],z_1,\cdots,z_L\}$ is generated from speech features $\boldsymbol{X}$ using a special token [dual] as

$$P(\boldsymbol{W}|\boldsymbol{X},[\mathrm{dual}];\boldsymbol{\Theta}_{\mathrm{joint}}) = \prod_{t=1}^{N+L+1} P(w_t|w_{1:t-1},\boldsymbol{X},[\mathrm{dual}];\boldsymbol{\Theta}_{\mathrm{joint}}), \quad (11)$$

where $\boldsymbol{\Theta}_{\mathrm{joint}}$ is the trainable parameter set. Also, in the ASR and S2WT tasks, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are generated from $\boldsymbol{X}$ using special tokens [spoken] and [written], respectively as

$$P(\boldsymbol{Y}|\boldsymbol{X},[\mathrm{spoken}];\boldsymbol{\Theta}_{\mathrm{joint}}) = \prod_{n=1}^{N} P(y_n|y_1,\cdots,y_{n-1},\boldsymbol{X},[\mathrm{spoken}];\boldsymbol{\Theta}_{\mathrm{joint}}), \quad (12)$$

$$P(\boldsymbol{Z}|\boldsymbol{X},[\mathrm{written}];\boldsymbol{\Theta}_{\mathrm{joint}}) = \prod_{l=1}^{L} P(z_l|z_1,\cdots,z_{l-1},\boldsymbol{X},[\mathrm{written}];\boldsymbol{\Theta}_{\mathrm{joint}}). \quad (13)$$

Training: The S2SWT model is trained in a two-stage manner. In the first stage, ASR, S2WT, and S2SWT tasks are trained simultaneously in a single model. When a special token [spoken] or [written] is set, the ASR or S2WT is trained. Also, when a special token [dual] is set, the S2SWT is trained. The loss function is defined as

$$\mathcal{L}_{\mathrm{joint}} = \mathcal{L}_{\mathrm{asr}} + \mathcal{L}_{\mathrm{s2wt}} + \mathcal{L}_{\mathrm{s2swt}}, \quad (14)$$

$$\mathcal{L}_{\mathrm{asr}} = -\sum_{(\boldsymbol{X},\boldsymbol{Y})\in\mathcal{D}_{\mathrm{asr}}} \log P(\boldsymbol{Y}|\boldsymbol{X},[\mathrm{spoken}];\boldsymbol{\Theta}_{\mathrm{joint}}), \quad (15)$$

$$\mathcal{L}_{\mathrm{s2wt}} = -\sum_{(\boldsymbol{X},\boldsymbol{Z})\in\mathcal{D}_{\mathrm{s2wt}}} \log P(\boldsymbol{Z}|\boldsymbol{X},[\mathrm{written}];\boldsymbol{\Theta}_{\mathrm{joint}}), \quad (16)$$

$$\mathcal{L}_{\mathrm{s2swt}} = -\sum_{(\boldsymbol{W},\boldsymbol{Z})\in\mathcal{D}_{\mathrm{s2swt}}} \log P(\boldsymbol{W}|\boldsymbol{X},[\mathrm{dual}];\boldsymbol{\Theta}_{\mathrm{joint}}), \quad (17)$$

where $\mathcal{D}_{\mathrm{s2swt}}$ is a dataset that has sets of speech, spoken text, and written text. In the second stage, the joint model is fine-tuned using the S2SWT task with Eq. (17).

Decoding: The decoding problem for S2SWT is defined as

$$\hat{\boldsymbol{W}} = \arg\max_{\boldsymbol{W}} P(\boldsymbol{W}|\boldsymbol{X},[\mathrm{dual}];\boldsymbol{\Theta}_{\mathrm{joint}}). \quad (18)$$

5. Experiments

5.1. Datasets

We experimented with Japanese ASR and S2WT tasks using a well-known Corpus of Spontaneous Japanse (CSJ) [15]. We divided the CSJ into training (518 h), validation (1.9 h), and two test (1.8 and 1.3 h) datasets. Although this dataset contains paired utterance-level audio signal and spoken text (manual transcription) data, it lacks written text. Thus, we use crowdsourcing to create written text using a part of spoken text, following the method in [12]. We prepared 417,406 paired the audio signal and transcription data, of which 125,539 also had written text. In the training data, the ASR task had 417,406, and the ST2WT, S2WT, and S2SWT tasks had 125,539 paired data. Also, the validation set had 1,292 paired data, and the two test sets had 1,272 and 1,385 each. Note that the validation and test sets contained both the spoken and the written text.

5.2. Setup

We compare the proposed method with the following baseline models: ST2WT, cascading, individual, and interactive learning [27] models. The cascading and individual models are described in Sec. 3. The S2WT model in the individual model is initialized by training on ASR data and then fine-tuned on S2WT data because the amount of data for the S2WT was small. Also, the interactive learning model was originally proposed for ASR and ST tasks to perform synchronously and interactively in a single model. The model is initialized by training on ASR and S2WT data and then fine-tuned on S2SWT data.

We introduced a transformer encoder-decoder model [28] for each model. For these models, the transformer blocks were

Table 1: *Example outputs of individual model and proposed method.*

	Spoken text	Written text
Reference	ん第百回というのは昭和二十八年	第１００回は昭和２８年の
Individual	ん第百回というのは昭和二十八年	第１０回というのは昭和２８年です。
Proposed	ん第百回というのは昭和二十八年	第１００回というのは昭和２８年です。
Translation	uh the 100th was in showa 28 years	The 100th was in showa 28 years.
Reference	でがとをがあー両極に来て	「が」と「を」が両極に来て、
Individual	でがとおがあー両極に来て	「が」と両極に来て、
Proposed	でがとおがあー両極に来て	「が」と「お」が両極に来て、
Translation	so the ga and wo are uh they come to both poles	The “ga” and “wo” come to both poles,

Table 2: *Results of ASR and S2WT tasks. Since ST2WT is trained from error-free spoken text, the training data is different from other methods.*

Method	Params	WER (↓)	BLEU (↑)
ST2WT	29.4M	–	0.596
Cascading	58.8M	6.2%	0.552
Individual	58.8M	6.2%	0.558
Interactive	33.6M	6.9%	0.550
Proposed	29.4M	**6.0%**	**0.564**

composed under the following conditions: the dimensions of the output continuous representations were set to 512, the dimensions of the inner outputs in the position-wise feed-forward networks were set to 1,024, and the number of heads in the multi-head attention was set to 4. In the nonlinear transformational function, the Swish activation was used. For an encoder, we used 80 log mel-scale filterbank coefficients as acoustic features. The frame shift was 10 ms. The acoustic features passed two convolution and max pooling layers with a stride of 2, so we down-sampled them to $1/4$ along with the time axis. After these layers, we stacked 6-layer transformer encoder blocks. For a decoder, we used 512-dimensional character embeddings where the vocabulary size was set to 3,316. We also stacked 4-layer transformer decoder blocks. For training, we used RAdam optimizer [29]. We set the mini-batch size to 64 utterances and the dropout rate in the transformer blocks to 0.1. We introduced label smoothing where its soothing parameter was set to 0.1 and applied SpecAugment [30]. Our SpecAugment only applied frequency masking and time masking. For testing, we used a beam search algorithm in which the beam size was set to 4. In the interactive learning model, we introduce an interactive attention sub-layer [27] into the transformer block in the decoder. Also, we set $\lambda = 0.3$, which is a hyper-parameter to control how much information of the other task should be taken into consideration [27] and $k = 3$ for wait-k policy [31]. As evaluation metrics, we calculated the word error rate (WER) for the ASR task and 4-gram BLEU [32] for the S2WT task.

5.3. Results

Table 1 shows examples of the output of the proposed method and the interactive model. Table 2 lists the results of each model along with their parameter sizes. The scores in the table were the average of each score calculated using two test data. The cascading and individual models have double the amount of parameters because they use two models. The interactive learning model has more parameters than the proposed method because it requires an additional attention layer.

First, we focus on the results of the conventional methods. Table 2 shows that the BLEU score of the individual model outperformed that of cascading. This indicates that optimizing written text directly from speech improves the performance of generating written text. On the other hand, the performance of the interactive learning model underperformed other methods. We think the amount of data was insufficient to learn the relationship between the tasks in our experiments. The interactive learning model trains the relationship between ASR and S2WT tasks using the additional cross-attention layer. Thus, it is supposed that a large set of speech, spoken text, and written text are required to output effective representation for each other.

Next, we focus on the result of the proposed method. Table 2 shows that the proposed method outperformed conventional methods. The improvement in the BLEU score suggests that the proposed method can train the relationship between spoken and written text by outputting spoken and written dual text. Actually, the individual model predicted written text based only on speech, tokens that could have output by ASR were omitted, and conversion errors (e.g., 100 was output as 10) occurred, as shown in Table 1. On the other hand, the proposed method can predict written text using both speech features and spoken text, so the errors that occurred in the individual model were improved. Thus, it is inferred that generating the spoken and written dual text from the speech is effective for the S2WT task. Also, the improvement in the WER score suggests that the two-stage learning in the proposed method was effective. To use paired speech and spoken text data, we trained ASR using the data in the first stage. In the second stage, the spoken text was trained from a small amount of speech, but the performance of ASR was maintained by using the parameters in the first stage. Therefore, we think the proposed method is an effective modeling method regarding computational cost and performance to output both the spoken and the written text.

6. Conclusion

In this paper, we proposed a speech-to-spoken and written text (S2SWT) that generates spoken and written dual text from speech features in an autoregressive model. The S2SWT model can consider the relationship between spoken and written text because the written text is generated from speech features and spoken text by outputting written text after spoken text. Also, we can reduce the computational cost by outputting both the spoken and the written text with one model. Thus, the proposed method is an effective modeling method from the viewpoint of computation complexity and performance. Our experimental results on ASR and S2WT tasks using the CSJ demonstrated that the proposed method outperformed conventional methods.

7. References

[1] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2016, pp. 4945–4949.

[2] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs rnn in speech applications," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 449–456.

[3] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, pp. 12 449–12 460, 2020.

[4] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, 2017, pp. 1073–1083.

[5] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *Proc. International Conference on Machine Learning (ICML)*, 2020, pp. 11 328–11 339.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[7] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, pp. 339–351, 2017.

[8] Y.-N. Chen, W. Y. Wang, and A. I. Rudnicky, "Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013, pp. 120–125.

[9] L. Qin, W. Che, Y. Li, H. Wen, and T. Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," in *Proc. Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 2078–2087.

[10] E. Shriberg, "To 'errrr'is human: ecology and acoustics of speech disfluencies," *Journal of the international phonetic association*, pp. 153–169, 2001.

[11] J. Liao, S. E. Eskimez, L. Lu, Y. Shi, M. Gong, L. Shou, H. Qu, and M. Zeng, "Improving readability for automatic speech recognition transcription," *Transactions on Asian and Low-Resource Language Information Processing*, 2020.

[12] M. Ihori, A. Takashima, and R. Masumura, "Parallel corpus for japanese spoken-to-written style conversion," in *Proc. the Twelfth Language Resources and Evaluation Conference (LERC)*, 2020, pp. 6346–6353.

[13] J. Nozaki, T. Kawahara, K. Ishizuka, and T. Hashimoto, "End-to-end speech-to-punctuated-text recognition," *arXiv e-prints*, pp. arXiv–2207, 2022.

[14] H. Futami, E. Tsunoo, K. Shibata, Y. Kashiwagi, T. Okuda, S. Arora, and S. Watanabe, "Streaming joint speech recognition and disfluency detection," *arXiv preprint arXiv:2211.08726*, 2022.

[15] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of japanese," in *Proc. International Conference on Language Resources and Evaluation (LREC)*, 2000, pp. 947–9520.

[16] N. Bach and F. Huang, "Noisy BiLSTM-Based Models for Disfluency Detection," in *Proc. Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 4230–4234.

[17] S. Wang, W. Che, Y. Zhang, M. Zhang, and T. Liu, "Transition-based disfluency detection using lstms," in *Proc. the Conference on Empirical Methods in Natural Language Processing (EMLP)*, 2017, pp. 2785–2794.

[18] Q. Dong, F. Wang, Z. Yang, W. Chen, S. Xu, and B. Xu, "Adapting translation models for transcript disfluency detection," in *Proc. of the Conference on Artificial Intelligence (AAAI)*, 2019, pp. 6351–6358.

[19] S. Kim, "Deep recurrent neural networks with layer-wise multi-head attentions for punctuation restoration," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7280–7284.

[20] O. Tilk and T. Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration." in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016, pp. 3047–3051.

[21] M. Á. Tündik, B. Tarjan, and G. Szaszák, "A bilingual comparison of maxent-and rnn-based punctuation restoration in speech transcripts," in *Proc. the International Conference on Cognitive Infocommunications (CogInfoCom)*, 2017, pp. 121–126.

[22] E. Pusateri, B. R. Ambati, E. Brooks, O. Platek, D. McAllaster, and V. Nagesha, "A mostly data-driven approach to inverse text normalization." in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 2784–2788.

[23] M. Sunkara, C. Shivade, S. Bodapati, and K. Kirchhoff, "Neural inverse text normalization," *arXiv preprint arXiv:2102.06380*, 2021.

[24] J. Guo, T. N. Sainath, and R. J. Weiss, "A spelling correction model for end-to-end speech recognition," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5651–5655.

[25] O. Hrinchuk, M. Popova, and B. Ginsburg, "Correction of automatic speech recognition with transformer sequence-to-sequence model," in *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7074–7078.

[26] H. Le, J. Pino, C. Wang, J. Gu, D. Schwab, and L. Besacier, "Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation," in *Proc. International Conference on Computational Linguistics (COLING)*, 2020, pp. 3520–3533.

[27] Y. Liu, J. Zhang, H. Xiong, L. Zhou, Z. He, H. Wu, H. Wang, and C. Zong, "Synchronous speech recognition and speech-to-text translation with interactive decoding," in *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 8417–8424.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances in neural information processing systems (NIPS)*, 2017, pp. 5998–6008.

[29] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *Proc. International Conference on Learning Representations (ICLR)*, 2020.

[30] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," pp. 2613–2617, 2019.

[31] M. Ma, L. Huang, H. Xiong, R. Zheng, K. Liu, B. Zheng, C. Zhang, Z. He, H. Liu, X. Li, H. Wu, and H. Wang, "STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Proc. Annual Meeting of the Association for Computational Linguistics (ACL), 2019, pp. 3025–3036.

[32] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. Annual Meeting on Association for Computational Linguistics (ACL)*, 2002, pp. 311–318.