Neural Speech Completion

Kazuki Tsunematsu[1], *Johanes Effendi*[1,2], *Sakriani Sakti*[1,2], *and Satoshi Nakamura*[1,2]

[1]Nara Institute of Science and Technology, Japan
[2]RIKEN, Center for Advanced Intelligence Project (AIP), Japan
{tsunematsu.kazuki.tj5,johanes.effendi.ix4, ssakti, s-nakamura}@is.naist.jp

Abstract

During a conversation, humans often predict the end of a sentence even when the other person has not finished it. In contrast, most current automatic speech recognition systems remain limited to passively recognizing what is being said. But applications like voice search, simultaneous speech translation, and spoken language communication may require a system that not only recognizes what has been said but also predicts what will be said. This paper proposes a speech completion system based on deep learning and discusses the construction in a text-to-text, speech-to-text, and speech-to-speech framework. We evaluate our system on domain-specific sentences with synthesized speech utterances that are only 25%, 50%, or 75% complete. Our proposed systems provide more natural suggestions than the Bidirectional Encoder Representations from Transformers (BERT) language representation model.

Index Terms: speech recognition, speech completion, sequence-to-sequence deep learning

1. Introduction

Spoken communication is a joint activity where the transformation of an intention in the speaker's mind into an understanding in the listener's mind occurs [1]. During a conversation, speakers construct grammatical sentences based on thoughts, execute them through vocal organs, and produce speech utterances. Listeners analyze the speech sounds and ultimately decode the utterances into meaning [2]. Research has found that our auditory system and brain play a decisively proactive role that anticipates a possible word or detects errors in the prediction [3, 4, 5, 6]. This capability helps us understand what is being said quickly and efficiently, especially in complicated, uncertain, and noisy conditions. Fig. 1 shows an example where an elderly speaker cannot remember an entire phrase and hesitates, and a listener helps by suggesting an option. Other studies also performed speaker-listener neural-coupling experiments and concluded that prediction is a critical aspect of successful communication [2, 7].

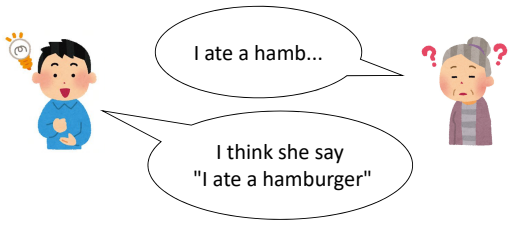

Figure 1: *Example of anticipation of possible words in human communication.*

In human-machine interactions, the machine can also perform completion automatically, which is a process that predicts the next word a user intends to enter after only a few characters have been typed into a text input field. The original idea of this function was to help individuals with physical disabilities [8], but it is currently widely used in many applications. Several search engines use predictive search to find and answer questions more quickly [9]. Since it could provide good predictions in domains with a limited number of possible words, many text editors and command-line interpreters (i.e., Emacs, C shell, or bash) utilize it to complete the names of files and commands. Code completion also helps people write code faster [10]. Recently, many of these applications use a deep learning framework to improve performance [11, 12, 13].

Despite extensive research works on autocomplete, it remains limited to text-based human-machine interaction. Most human-machine speech communications rely on automatic speech recognition (ASR) as their interface. Researchers have been working on speech recognition technology for decades, and several approaches have been proposed, including template-based with dynamic time warping (DTW) [14, 15] and statistical modeling of hidden Markov model-Gaussian mixture model (HMM-GMM) [16, 17]. Deep learning has become the mainstream, and the latest ASR systems have almost achieved human parity in the switchboard conversational speech recognition task [18, 19]. Nevertheless, most ASR systems remains limited to the passive recognition of what is being said without the capability to predict what will be said.

Although research has addressed speech completion tasks, it is unfortunately limited. Goto et al. [20] is one of the few studies that attempted it with a speech recognizer that was extended with a vocabulary tree. Candidates of the complete text transcriptions were obtained by tracing the vocabulary tree from the top of the ASR hypothesis to its leaves. However, it is based on the traditional HMM-based ASR framework and is only applied to the speech-to-text task of a search engine. Recently, Niehues et al. [21, 22] discussed simultaneous speech interpretation where a translation system is required to provide an initial translation in real-time before the complete sentence has actually been spoken. The work presented the possibility of translating partial sentences of the source language into complete sentences of the target language. Such problems could be handled on the ASR by completing the sentences of the source language before the text-translation starts.

Since applications like voice search, simultaneous speech translation, and spoken language communication generally require a system that can recognize what has been mentioned and predict what will be said, we propose a speech completion system based on deep learning. We discuss various possibilities for neural completion systems, including construction in text-to-text, speech-to-text, and speech-to-speech frameworks. We evaluate our system on domain-specific sentences with synthesized speech utterances that are only 25%, 50%, or 75% complete. To the best of our knowledge, this is the first work that develops an end-to-end neural speech-to-speech completion task.

2. Overview of Proposed Framework

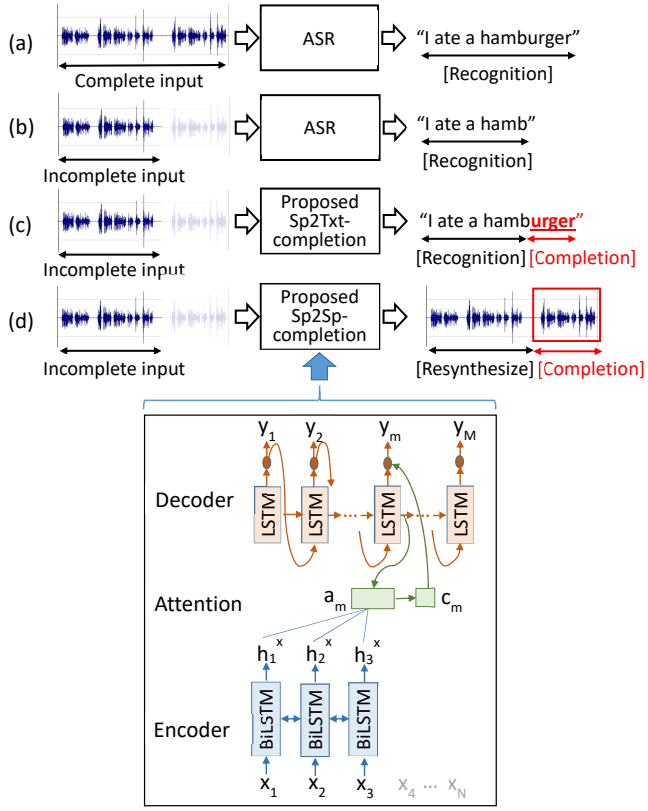

Figure 2: *Overview of proposed sequence-to-sequence completion system in comparison with standard ASR system: (a) ASR system given a complete speech utterances; (b) ASR system given a incomplete speech utterances; (c) proposed speech-to-text completion system given incomplete speech utterances; (d) proposed speech-to-speech completion system given incomplete speech utterances.*

Figures 2(a) and (b) illustrate the ASR system given complete and incomplete speech utterances, and Fig. 2(c) and (d) show the proposed speech-to-text and speech-to-speech framework, respectively, given incomplete speech utterances. The speech-to-text completion system provides not only the text recognition result but also an extension to complete the text, while the speech-to-speech completion system attempts to produce a complete speech utterance. We based our proposed completion system on the sequence-to-sequence attention-based neural network architecture [23, 24]. The details are discussed below.

2.1. Sequence-to-sequence framework given partial input sequences

In a standard sequence-to-sequence framework, given complete input sequences $\mathbf{x} = [x_1, x_2, ..., x_n, ..., x_N]$ with length N, we directly model conditional probability $p(\hat{\mathbf{y}}|\mathbf{x})$ and output corresponding sequences $\hat{\mathbf{y}} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_n, ..., \hat{y}_M]$ with length M. In contrast, in this framework, the system only receives partial input sequences $\mathbf{x} = [x_1, x_2, ..., x_P]$ with length P where $P < N$ (see the example in Fig. 2 bottom side where $P = 3$).

The overall structure of the attention-based encoder-decoder model consists of encoder, decoder, and attention modules. The encoder uses a bidirectional recurrent neural network with long short-term memory (bi-LSTM) units and produces sequence of vector representation $h^x = (h_1^x, h_2^x, ..., h_P^x)$. The forward LSTM reads the input sequence from x_1 to x_P and estimates forward $\overrightarrow{h^x}$, and the backward LSTM reads the input sequence in reverse order from x_P to x_1 and estimates backward $\overleftarrow{h^x}$. Thus, for each input x_p we obtain h_p^x by concatenating forward $\overrightarrow{h^x}$ and backward $\overleftarrow{h^x}$.

The attention module estimates context information c_t of the incomplete input sequence over encoder hidden states h_p^x:

$$c_m = \sum_{p=1}^{P} a_m(p) * h_p^x, \tag{1}$$

$$a_m(p) = \text{Align}(h_p^x, h_m^{\hat{y}}) = \frac{\exp(\text{Score}(h_p^x, h_m^{\hat{y}}))}{\sum_{s=1}^{S} \exp(\text{Score}(h_p^x, h_m^{\hat{y}}))}. \tag{2}$$

There are several variations for score functions [25]:

$$\text{Score}(h_p^x, h_m^{\hat{y}}) = \begin{cases} \langle h_p^x, h_m^{\hat{y}} \rangle, & \text{dot product} \\ h_p^{e\intercal} W_p h_m^{\hat{y}}, & \text{bilinear} \\ V_p^\intercal \tanh(W_p[h_p^x, h_m^{\hat{y}}]), & \text{MLP.} \end{cases} \tag{3}$$

Decoder hidden activation vector $\tilde{h}_m^{\hat{y}}$ is computed by applying linear layer W_c over context information c_t and current hidden state $h_m^{\hat{y}}$,

$$\tilde{h}_m^{\hat{y}} = \tanh(W_c[c_m; h_m^{\hat{y}}]). \tag{4}$$

After that, with the uni-directional LSTM (forward only) and complete output $\hat{y}_1, \hat{y}_2, ..., \hat{y}_n, ..., \hat{y}_M$ is predicted, based on the whole sequence of the previous output:

$$p(\hat{y}_m|\hat{y}_1, \hat{y}_2, ..., \hat{y}_{m-1}, x) = softmax(W_y \tilde{h}_m^{\hat{y}}). \tag{5}$$

2.2. Text-to-text completion system

This framework resembles attention-based machine translation (MT) [23]. In this case, input sequence $\mathbf{y} = [y_1, ..., y_Q]$ is a incomplete text transcription, and target sequence $\hat{\mathbf{y}} = [\hat{y}_1, ..., \hat{y}_M]$ is the predicted complete transcription, where $Q < M$.

2.3. Speech-to-text completion system

This framework resembles the attention-based ASR [26]. In this case, input sequence $\mathbf{x} = [x_1, ..., x_P]$ is incomplete speech utterances, and target sequence $\hat{\mathbf{y}} = [\hat{y}_1, ..., \hat{y}_M]$ is the estimated completed transcription (see Fig. 2(c)).

2.4. Speech-to-speech completion system

In this framework, input sequence $\mathbf{x} = [x_1, ..., x_P]$ is the incomplete speech utterances, and target sequence $\hat{\mathbf{x}} = [\hat{x}_1, ..., \hat{x}_N]$ is the estimated completed speech utterances, and $P < N$ (see Fig. 2(d)). The encoder has similar architecture to the speech-to-text encoder, and the decoder has similar architecture to the text-to-speech decoder. Here, we pre-train the encoder on the speech-to-text completion task resembles to the one mentioned in Sec. 2.3, and the decoder on the text-to-speech completion task resembles the attention-based text-to-speech synthesis (TTS) [27]. After initialization with pre-trained components, the speech-to-speech completion system is fine-tuned

in an end-to-end fashion. We model the conditional probability between $p(\hat{x}|y)$, where $y = [y_1, ..., y_Q]$ is the partial-length sequence of characters with length Q and $\hat{x} = [\hat{x}_1, ..., \hat{x}_N]$ is the full-length sequence of speech features with frame-length N.

3. Experimental Set-up

3.1. Text and Speech Corpus

Since speech-to-speech completion systems need to complete the speech utterances with speech characteristics that are identical as the speech input, we limit our work to single-speaker data. However, because obtaining a large amount of single-speaker speech data is challenging, we synthesized the speech with Google TTS[1].

The text material was based on the Basic Travel Expression Corpus (BTEC) [28, 29], that covers basic conversations in the travel domain. The data was chosen to reduce the high uncertainty in the completion process and focus on a specific domain. The training, development, and test set configurations can be found in Table 1. We converted all of the sentences into lowercase letters and removed all the punctuation marks [,:?.]. We used 26 letters (a-z) and three special tags (<s>, </s>, <spc>) that denote the start and the end of sentences and the spaces between words.

Table 1: *BTEC Dataset.*

Corpus	Number of sentences		
	Train	Val	Test
BTEC	157448	4870	510

Next we created incomplete data in which we cut the full set of text or speech utterances into 25%, 50%, and 75% partial-lengths from the generated synthesized speech and the corresponding transcription.

3.2. Feature Extraction

All the raw speech waveforms were represented at a 16-kHz sampling rate. For the speech features, we used a log magnitude spectrogram extracted by the short-time Fourier transform (STFT) from the Librosa library[2]. First, we applied wave-normalization (scaling raw wave signals into range [-1, 1]) per utterance, followed by pre-emphasis (0.97), and extracted the spectrogram with an STFT, a 50-ms frame length, a 12.5-ms frame shift, and a 2048 point FFT. We transformed all of the speech utterances into log-scale and normalized each feature into 0 mean and unit variances. Our final set included 80-dimension log Mel-spectrogram features and 1025-dimension log magnitude spectrograms.

3.3. Baseline Systems

For comparison, we used two baseline systems: (1) recurrent neural network language model (RNN-LM) and (2) BERT.

3.3.1. RNN-LM

RNN-LM is a language model that operates a predictive model for the next token, given the previous ones and the state of the hidden layer in the previous time step [30]. The input layer uses the 1-of-N representation of previous token w_{i-1} concatenated with the previous state of hidden layer h_{i-1}. The neurons in hidden layer h_i use the sigmoid activation function. The output layer estimates the probability distribution of the next word given the history and output w_i. In this study, the length of the output was not determined. The RNN-LM prediction is performed repeatedly until the [EOS] token appears that indicates the end of the sentence.

3.3.2. BERT

A traditional LM is based on a single-directional (left-to-right) approach that predicts the next word given a sequence. Unfortunately, such an approach limits context learning. Bidirectional Encoder Representations from Transformers (BERT) is a "language understanding" model [31] that is bi-directionally (left-to-right and right-to-left) trained on a massive text corpus. In contrast with a traditional LM, BERT has a more profound sense of language context. BERT exploits a transformer [32], which is an attention mechanism that bi-directionally learns the contextual relations between words (or sub-words) in a text. There are two types of pre-learning tasks in BERT: a masked language model and subsequent sentence prediction. We only utilized a pre-trained BERT that leveraged the masked language model (Masked LM).

The masked language model is a task that replaces part of the input sequence with [MASK] tokens and predicts it. In this study, instead of randomly replacing [MASK] in the middle of the sentences, we replaced the incomplete part at their end with a [MASK] token for prediction. But since the number of tokens is unknown that has to be predicted, the number of [MASK] tokens is set to be identical as the maximum sentence length in the data. In this research, since the classification task and the following sentence are not predicted, we did not use the [CLS] token for class embedding or the [SEP] token for separating sentences. The sentences are simply divided for each input.

3.4. Proposed Systems

Our attention-based encoder-decoder model used three stacked BiLSTM encoders, a single layer LSTM, and multilayer perceptron (MLP)-based attention [25] components. The log-scaled Mel-spectrogram were fed into a fully connected layer and transformed by a LeakyReLU ($l = 1e - 2$) [33] activation function. This model doesn't need a language model or a word dictionary. As for a speech-to-speech completion system, the speech decoder is based on a sequence-to-sequence TTS (Tacotron) [27], but we changed the GRU into two stacked LSTMs with 256 hidden units. Further details of our ASR and TTS parameter set-up are available here [34].

4. Experiment Results

We performed our text-to-text, speech-to-text, and speech-to-speech completion system and evaluated it on both word and sentence completion tasks. Table 2 shows an example of word and sentence completion. We calculated the character or word error rate (CER/WER) by comparing the completed sentence hypothesis with the ground truth sentence.

Table 2: *Example of word and sentence completion.*

Input	I ate a hamb
Word completion	I ate a hamb**urger**
Sentence completion	I ate a hamb**urger at a restaurant**

4.1. Text-to-text completion

We conducted word completion evaluations and investigated the performance of our text-to-text completion system. Given a word sequence of speech utterance, where the last word was incomplete, the system needed to produce complete word. Although there might be additional words needed to complete the

[1]https://pypi.python.org/pypi/gTTS

[2]Librosa–https://librosa.github.io/librosa/0.5.0/index.html

sentence, this task only focus on completing the partial word. For comparison, we also asked 15 subjects whose TOEIC (The Test of English for International Communication) scores exceeded 730 or higher to do the same. The results are shown in Table 3. The performance of the proposed method surpassed human completion even when compared with the best results among all subjects.

Table 3: *Performance of text-to-text word completion system in CER (%).*

	CER (%)
Proposed system	2.70
Human	7.21
Human (best)	5.50

4.2. Speech-to-text completion

Next, we investigated the performance of our speech-to-text system on sentence completion. As a comparison, we also included our text-to-text sentence completion system and the baseline RNN-LM and BERT. Given speech utterances that only are 25%, 50%, or 75% complete, the system needed to produce complete sentences.

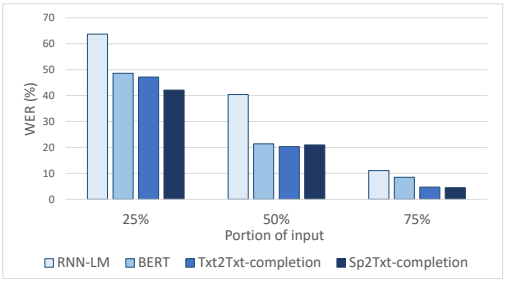

Figure 3: *Performance of sentence completion system in terms of WER.*

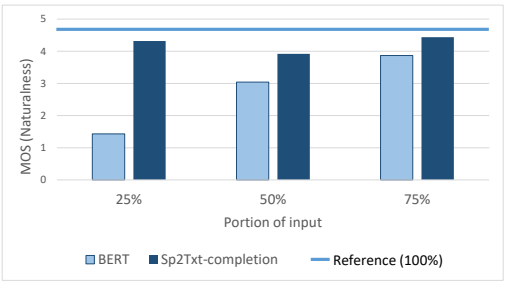

Figure 4: *Performance of sentence completion system in terms of naturalness.*

The completion performance in the WER is depicted in Fig. 3. We expected that the shorter the input speech, the more challenging it would be to produce a complete sentence. Overall, the proposed text-to-text, speech-to-text completion system outperformed the baseline RNN-LM and BERT. Although BERT is robust for filling some missing parts, it seems weak for filling long missing parts that locates at the end of the sentence because there are no right contexts that can be used to leverage prediction of the missing parts.

Since the system might suggest a complete sentence that may be different from the reference, we also subjectively evaluated the mean opinion score (MOS) [35] tests. 12 subjects participated whose ages ranged from 20 to 40 with TOEIC scores of 730 or higher. They read each presented text and rated its naturalness on a 5-point MOS scale, where 5 indicated 'completely natural' and 1 indicated 'completely unnatural.' The MOS results are presented in Fig. 4. The results also reveal that our proposed completion system provided more natural suggestions than the BERT language representation model.

4.3. Speech-to-speech completion

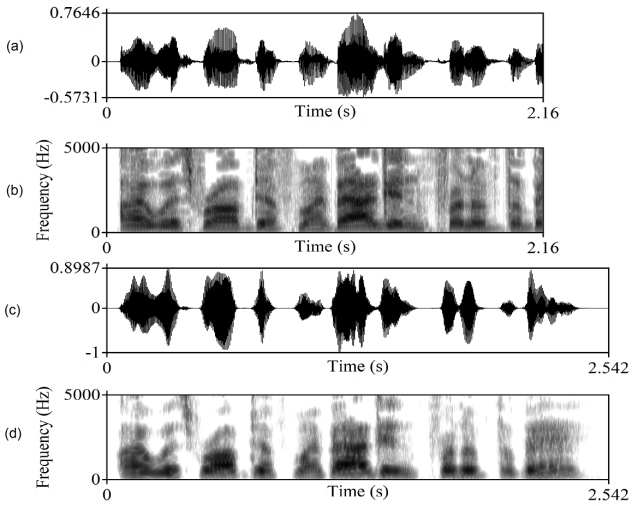

Figure 5: *Waveform and spectrogram comparisons of partial input speech (top: (a) and (b)) and generated speech completion (bottom: (c) and (d)).*

Figure 5 shows an example of the waveform and spectrogram comparisons of the partial input speech (top: (a) and (b)) and the generated speech completion (bottom: (c) and (d)) for utterance "I was robbed of my bag in front of the b," which becomes "I was robbed of my bag in front of the bank." The system resynthesized the speech input and completed the speech while retaining the characteristics of the speech style. For further information on several speech-to-speech samples, see the following reference: https://sites.google.com/ahclab.naist.jp/neuralspeechcompletion/home.

5. Conclusions

We described the possibility of utilizing a sequence-to-sequence deep learning framework for a new task: word- and sentence-based speech completion. Our proposed system provided complete full-length results closer to the reference transcriptions (with a lower error rate) and more natural suggestions than the baseline. The speech-to-speech completion system resynthesized the speech input and completed the remaining part while retaining the characteristics of the speaker's speech style. Such a simple yet efficient approach enables people to easily reproduce these works. In the future, we will refine our framework and incorporate it within an incremental ASR and further investigate the capability using multi-speaker natural speech data.

6. Acknowledgements

Part of this work is supported by JSPS KAKENHI Grant Numbers JP17H06101 and JP17K00237.

7. References

[1] P. Denes and E. Pinson, *The Speech Chain*, ser. Anchor books. Worth Publishers, 1993. [Online]. Available: https://books.google.co.jp/books?id=ZMTm3nlDfroC

[2] G. J. Stephens, L. J. Silbert, and U. Hasson, "Speaker–listener neural coupling underlies successful communication," *Proceedings of the National Academy of Sciences*, vol. 107, no. 32, pp. 14 425–14 430, 2010.

[3] J. M. Honeycutt, M. L. Knapp, and W. G. Powers, "On knowing others and predicting what they say," *Western Journal of Speech Communication*, vol. 47, no. 2, pp. 157–174, 1983.

[4] S. Dikker, L. J. Silbert, U. Hasson, and J. D. Zevin, "On the same wavelength: predictable language enhances speaker-listener brain-to-brain synchrony in posterior superior temporal gyrus," *The Journal of Neuroscience*, vol. 34, no. 18, pp. 6267–6272, 2014.

[5] N. Pitrelli, D. Ramani, and A. Tavecchio, "I predict your words: that is how we understand what others say to us," *EurekAlert! the American Association for the Advancement of Science*, 2019.

[6] Y. Vidal, P. Brusini, M. Bonfieni, J. Mehler, and T. A. Bekinschtein, "Neural signal to violations of abstract rules using speech-like stimuli," *eNeuro*, vol. 6, no. 5, pp. 1–14, 2019.

[7] U. Hasson, A. A. Ghazanfar, B. Galantucci, S. Garrod, and C. Keysers, "Brain-to-brain coupling: A mechanism for creating and sharing a social world," *Trends Cognitive Science*, vol. 16, no. 2, p. 114–121, 2012.

[8] C. Tam and D. Wells, "Evaluating the benefits of displaying word prediction lists on a personal digital assistant at the keyboard level," *Assistive Technology*, vol. 21, no. 3, p. 105–114, 2009.

[9] S. Hollingsworth, "Google autocomplete: A complete SEO guide," *Search Engine Journal*, 2018.

[10] V. Raychev, M. T. Vechev, and E. Yahav, "Code completion with statistical language models," in *Proceedings of the 38th International ACM SIGPLAN Conference*, 2014.

[11] D. H. Park and R. Chiba, "A neural language model for query auto-completion," in *Proceedings of the 38th International ACM SIGIR Conference*, 2017.

[12] A. Bulana and I. Khorlo, "A neural net brain for an autocompletion engine: Improving the ux through machine learning," in *Proceedings of the SAS Global Forum*, 2019.

[13] TabNine, "Autocompletion with deep learning," https://www.tabnine.com/blog/deep/, 2019.

[14] T. K. Vintsyuk, "Speech discrimination by dynamic programming," *Kibernetika*, pp. 81–88, 1968.

[15] H. Sakoe and S. Chiba, "Dynamic programming algorithm quantization for spoken word recognition," *IEEE Transaction on Acoustics, Speech and Signal Processing*, vol. ASSP-26, no. 1, pp. 43–49, 1978.

[16] F. Jelinek, "Continuous speech recognition by statistical methods," *IEEE*, vol. 64, pp. 532–536, 1976.

[17] J. G. Wilpon, L. R. Rabiner, C. H. Lee, and E. R. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Transaction on Acoustics, Speech and Signal Processing*, vol. 38, no. 11, pp. 1870–1878, 1990.

[18] W. Xiong, J. Droppo, X. Huang, F. Seide, M. L. Seltzer, A. Stolcke, D. Yu, and G. G. Zweig, "Toward human parity in conversational speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 25, no. 12, pp. 2410–2423, 2017.

[19] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines," in *Procedings of the INTERSPEECH*, 2017, pp. 132–136.

[20] M. Goto, K. Itou, and S. Hayamizu, "Speech completion: On-demand completion assistance using filled pauses for speech input interfaces," in *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 2002, pp. 1489–1492.

[21] J. Niehues, T. S. Nguyen, E. Cho, T.-L. Ha, K. Kilgour, M. Muller, M. Sperber, S. Stueker, and A. Waibel, "Dynamic transcription for low-latency speech translation," in *Proceedings of the INTERSPEECH*, 2016, p. 2513–2517.

[22] J. Niehues, N. Pham, T. Ha, M. Sperber, and A. Waibel, "Low-latency neural speech translation," *arXiv preprint arXiv:1808.00491*, 2018.

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

[24] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2014, pp. 3104–3112.

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

[26] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[27] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: A fully end-to-end text-to-speech synthesis model," *arXiv preprint arXiv:1703.10135*, 2017.

[28] G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto, "Creating corpora for speech-to-speech translation," in *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH)*, 2003.

[29] G. Kikui, S. Yamamoto, T. Takezawa, and E. Sumita, "Comparative study on corpora for speech translation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1674–1682, 2006.

[30] S. Kombrink, T. Mikolov, M. Karafiát, and L. Burget, "Recurrent neural network based language modeling in meeting recognition," 2011, pp. 2877–2880.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2014.

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

[33] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[34] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 301–308.

[35] CCITT, *Absolute category rating (ACR) method for subjective testing of digital processors*. Red Book, 1984.