A Transformer-Based End-to-End Automatic Speech Recognition Algorithm

Fang Dong, Yiyang Qian, Tianlei Wang, Peng Liu, and Jiuwen Cao, *Senior Member, IEEE*

***Abstract*—End-to-End (E2E) automatic speech recognition (ASR) becomes popular recent years and has been widely used in many applications. However, current ASR algorithms are usually less effective when applied in specific applications with terminologies such as medical and economic fields. To address this issue, we propose a powerful Transformer based ASR decoding method for beam searching, called soft beam pruning algorithm (SBPA). SBPA can dynamically adjust the width of beam search. Meanwhile, a prefix module (PM) is added to access the contextual information and avoid removing professional words in the beam search. Combining SBPA and PM, the proposed ASR can achieve promising recognition performance on professional terminologies. To verify the effectiveness, experiments are conducted on real-world conversation data with medical terminology. It is shown that the proposed ASR achieved significant performance on both professional and regular words.**

***Index Terms*—Automatic speech recognition, soft beam pruning, prefix module, transformer, professional terminology.**

I. Introduction

AUTOMATIC speech recognition (ASR) becomes a major input system for several domains that need to deal with a large amount of repetitive transcription work, such as medical advice and telephone records. Compared to the conventional Hybrid hidden Markov model (HMM) based ASR [1], [2], the end-to-end (E2E) ASR systems that address the speech-to-text conversion problem with a single sequence-to-sequence model provide the state-of-the-art (SOTA) results, such as the Connectionist Temporal Classification (CTC) [3], [4] and Recurrent Neural Network transducer (RNN-T) [5], [6]. Specially, the recent attention-based encoder-decoder architectures (e.g., Transformer) [7], [8] become dominant in E2E ASR community.

Nevertheless, these existing ASR methods are generally designed for daily dialogues and trained on general speech dataset such as the popular Aishell dataset [9], [10]. Due to the difference of corpus co-occurrence distribution across multi-scenarios, these general ASR methods may suffer from performance degradation when applied to specific domain with terminologies, such as medical and financial ASR task. Particularly, these ASR methods fail to learn the regularities and grammar structures of terminologies when trained only on the general ASR dataset. Constructing a new dataset for the professional field is the direct way. For instance, Edwards et al. [11] and Chiu et al. [12] constructed medical speech data with 270 hours and 14,000 hours respectively for ASR training in medical fields. But these methods are generally time-consuming in constructing the domain-specified ASR datasets and it is difficult to generalize more different domains.

Another popular way is to refine the language model (LM) or decoding process in ASR. Mani et al. [13] proposed to learn a mapping from out-of-domain errors to in-domain medical terms. Zhao et al. [14] explored the shallow-fusion between independently trained LAS (Listen, Attend and Spell) and contextual n-gram models in beam search. Pundak et al. [15] introduced the shallow-fusion into RNN-T [6]. Jung et al. [16] improved the probabilities of given terminologies in a beam search based on acoustic model predictions without training. Kim et al. [17] used a word-matching algorithm with the backward search to overcome the limitations of contextual information recognition. However, these methods usually require addition modules to be trained to exploit the bias information such as the bias encoder [18], bias LM [19], class-based LM [20].

Compared to the aforementioned ways, the hot word customization is a more flexible and effective solution [21], [22]. To address these issues, a novel Transformer-based E2E ASR algorithm is developed using a list of terminology and prefix words in this letter. Specially, two modules including the soft beam pruning algorithm (SBPA) and the prefix word module (PM) are developed to improve the accuracy on the terminologies. The terminologies are weighted and enhanced by the SBPA module. The PM module is then applied to avoid beam search from pruning the terminology at the beginning. The beam search width is expanded by adding a new path if the generated word is similar to terminology after pruning. The main contributions of this study include

Manuscript received 13 May 2023; revised 17 September 2023; accepted 16 October 2023. Date of publication 27 October 2023; date of current version 9 November 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFE0205400 and Grant 2021YFE0100100; in part by the National Natural Science Foundation of China under Grant U1909209; and in part by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant 2022C03117. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sandro Cumani. *(Fang Dong and Yiyang Qian contributed equally to this work.) (Corresponding author: Jiuwen Cao.)*

Fang Dong is with the School of Information and Electrical Engineering, Hangzhou City University, Hangzhou 310015, China (e-mail: dongf@hzcu.edu.cn).

Yiyang Qian, Tianlei Wang, and Jiuwen Cao are with the Machine Learning and I-health International Cooperation Base of Zhejiang Province, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: qyy1010@hdu.edu.cn; tianlei.wang.cn@gmail.com; jwcao@hdu.edu.cn).

Peng Liu is with Zhejiang Baiying Technology Ltd. Company, Zhejiang 311100, China (e-mail: tianjin@byai.com).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LSP.2023.3328238, provided by the authors.

Digital Object Identifier 10.1109/LSP.2023.3328238

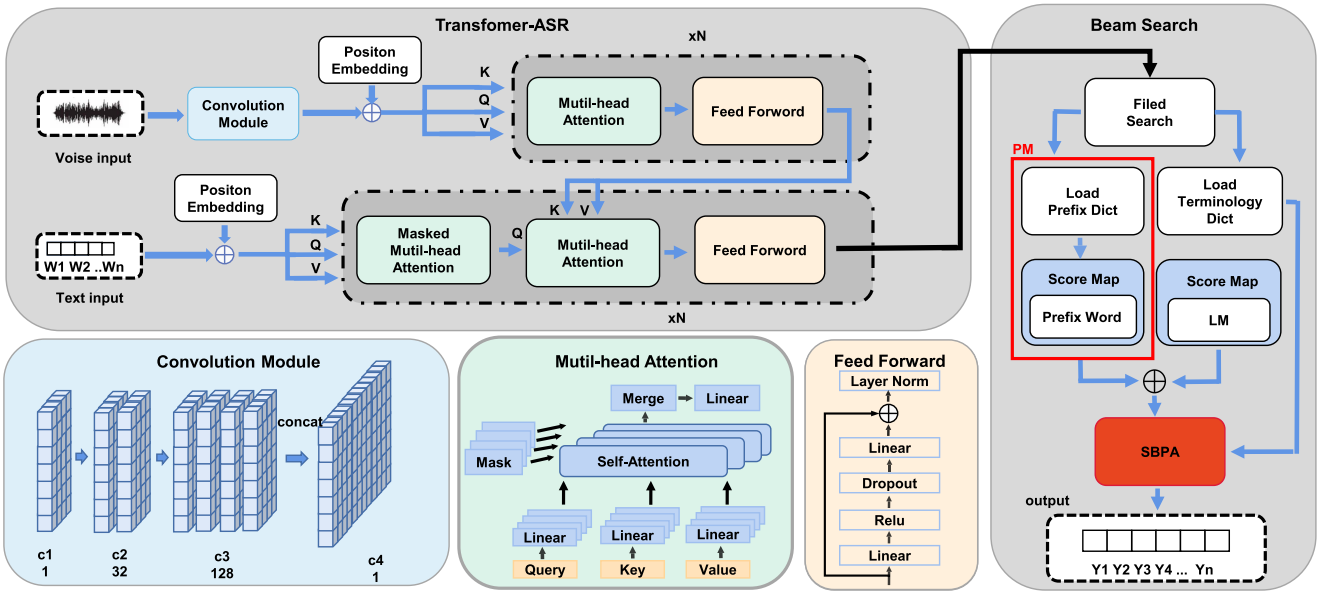


Fig. 1. Components of the whole speech transcription system

- The SBPA module is developed to promote the professional terminology recognition performance.
- The PM module of terminologies is proposed to avoid pruning terminology by the beam search at the beginning.
- A medical terminology dataset is built for performance evaluation with specialized terminologies from different departments, diseases and drugs.

Experimental results show that our method improves the ASR accuracy on the professional terminology recognition with the lowest Character Error Rate (CER) of 9.43%.

II. Proposed Transformer-based ASR Algorithm

The proposed Transformer-based ASR algorithm for specific domain with terminologies is presented in this section. As depicted in Fig. 1, the SBPA and PM modules are the main differences that will be described in the following.

A. Transformer-based ASR

Transformer is originally proposed in [7] and has been the dominant method in ASR community [8]. Transformer has advantages over existing systems in terms of word accuracy and training efficiency. Different from the translation tasks with complete word as tokens, the speech inputs are generally divided into incomplete words and phonemes to input the encoder of Transformer in ASR. Thus, similar to [23], a convolution module is used to merge more contextual information before the positional embedding layer. In addition, performing convolution allows for reducing the input size and increasing the number of channels at each stage. The convolution layer subsampling module concatenates all channels to one channel to induce multi-head attention.

In the network, 3 convolution layers with the kernel size of 3×3, the stride of s $= 2$, p padding (to deal with boundary conditions) are used. The channel sizes C_i are 1, 32 and 128. The height H_i and width W_i of the new token vector $X_i \in \mathbb{R}^{H_i \times W_i \times C_i}$ are:

$$H_i = \left[\frac{H_{i-1} + 2p - s}{sd} + 1\right], W_i = \left[\frac{W_{i-1} + 2p - s}{sd} + 1\right], \tag{1}$$

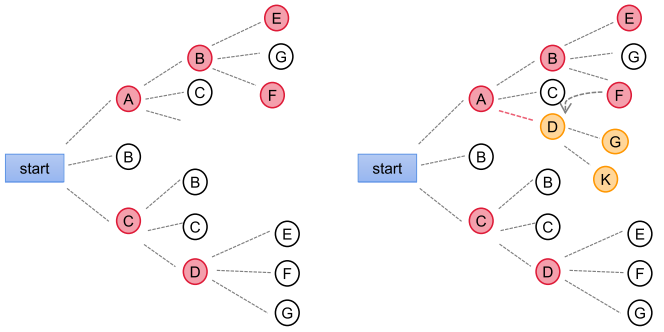
where sd is the sliding step in convolution layers.

In the decoder, a hybrid CTC/beam search method is used to improve the decoding accuracy. CTC learns to map the input and output to same length. The combination of CTC/beam search allows the time constraints of CTC to be incorporated in the attention by reweighting [24]. CTC is applied to generate the n best candidates, which are then re-scored on the beam search using the corresponding encoder output [25].

B. SBPA Module

Beam search is a common ASR decoding strategy for text generation tasks and speech recognition like CTC and Transformer. With beam search, the optimal solution in a relatively limited search space is found with less cost. Let X denote the sequence input, which is mapped to a vector representation in decoder, each X is paired with a target sentence Y, represents a response generation or a target sentence in encoder-decoder structure used in Transformer structure. $Y = \{Y_1, Y_2, ..., Y_i\}$ consists a sequence of i words. The distribution in text output is defined by the decoder layer, which predicts the tokens sequentially by the softmax functions.

As shown in Fig. 2(a), the beam search keeps K nodes (beam size) with scores of $S = \log P(y_1, y_1, ..., y_{t-1}|X)$. After each autoregression, K new nodes are constructed and then top K candidate nodes are selected for next autoregression. But only selecting the top K candidate nodes may ignore many correct words, especially for specific applications with terminologies. To address this problem, we propose a soft beam pruning algorithm in this letter.



(a) beam search (b) Improved soft beam pruning

Fig. 2. Schematic representation between beam search (left) and the proposed soft beam pruning algorithm (right).

For each $s \in V$, let $P_{AM}(s)$ be the probability of occurrence of token s based on an ASR, where V is the set of tokens. For a certain application with a prepared terminology dict Ter_{list}, we can decode the acoustic model output using a soft beam pruning algorithm. Each beam node $n_{t,k}$ has a state $s_{t,k}$. At same time, the node $n_{t,k}$ of the beam search finds the next K node from $n_{t+1,1}$ to $n_{t+1,K}$ according to the score. Along the beam search traverse, if the $s_{t,k}$ is similar to the last word in the term list terminology Ter_{list} with a of length J, it will be determined whether the complete word is similar to the terminology. Then, J nodes backward from $n_{t-1,new}$ to $n_{t-J,new}$ will be generated, and the algorithm will traverse $n_{t-J,new}$'s parents node in the reverse direction. A new path to $n_{t,new}$ will be created.

In current study, a new hyperparemeter terminology weight $W_{t,k}$, is proposed to control the strength. Large weights are assigned to the newly generated node which continues to the next round keyword nodes. For the node $n_{t-j,new}$, the terminology score is

$$W_{t,k}(s) = \begin{cases} w_j, & n_{t-j,new},\ j = 0, 1, ..., J \\ 0, & otherwise \end{cases}. \tag{2}$$

For each new node, the top $\widetilde{K}$ candidate nodes are selected according to the score, lending to the construction of $\widetilde{K} \times \widetilde{K}$ new nodes. Here, $\widetilde{K}$ is usually smaller than K by considering the arithmetic efficiency. In the beam search, the path between $n_{t-J-1,new}$ will be re-scored and compared with other path at t. Algorithm 1 summarizes the detailed soft beam pruning algorithm. Particularly, the beam search maximizes $\mathcal{S}$ to find candidate nodes as

$$\mathcal{S} = \log P_{AM}(s) + w_{LM} \log P_{LM} + \log W_{t,k}(s) \tag{3}$$

where $\log P_{AM}(s)$ and $\log P_{LM}$ represent the outputs of acoustic and language models, respectively. w_{LM} is the weight. The drawback is that the beam search may not provide more possibilities for a given state, which may promote a term that does not actually appear.

Algorithm 1: Soft Beam Pruning Algorithm.

$t \leftarrow t-1, \forall s \in V, \forall k \in K$
for $k \leq K$ **do**
  Step 1 : Generate a node backward
  if $s_{t,k} \in Ter_{list}(i, J)$ **then**
    while $j < Ter_{list}(i).length$ **do**
      if $s_{t-j,k} \notin Ter_{list}(i, j)$ **then**
        break
      end if
    end while
    $GenerateFlag = true$
  end if
  if $GenerateFlag = true$ **then**
    while $j < Ter_{list}(i).length$ **do**
      Generate($n_{t-j,new}$)
      $S_{t-j,new} = Ter_{list}(i, J-j)$
      $n_{t-j,new} \leftarrow traverse(n_{t-j+1,k})$
    end while
  end if
  Step 2 : Define the value of $W_{t,k}$
  if $n_{t,k} = n_0$ **then**
    $W_{t,k} \leftarrow 0\ \forall\ s \in V$
  else
    $W_{t,k} \leftarrow w_k\ \forall\ s \in children(n_{t,new})$
  end if
  Step 3 : New beam Search
  for j < J **do**
    $Generate(n_{t-j,\widetilde{k}})\ \widetilde{k} = 1, 2, .., \widetilde{K} \times \widetilde{K}$
    Calculate $Score_{t-j,\widetilde{k}}$ by (3)
    select top K score nodes in time step $t-j$
  end for
end for

C. PM Module

Soft beam pruning algorithm will clip out terminology with a rare first word of the term, which is usually generated after beam search and word generation. The PM module will avoid beam search to prune a specific term by increasing the weight between the term and its prefix word. PM module may reduce the recognition quality of sentences that do not contain any biased phrase.

In this letter, a biasing phrase is activated if it is proceeded by a set of common prefixes. For example, 'cold' is often used as the prefix of 'compress'. The weight between term and term's prefix word is increased. Particularly, a bias layer is applied to fused prefix word before the beam search pruning. A large number of unnecessary words matching the beam search and drown out the beam would be a problem. Overall, a certain number of phrases with activated prefix effect will be saved into the prefix dict. The speech transcription system will load the prefix dict in the beam search. At each t, when new output labels are generated by the decoder, the word will be decided whether belonging to a certain prefix dict or not. A small biasing weight λ_{Pre} will be assigned to the node. The beam search will find k other nodes to maximize

$$\mathcal{S} = \log P_{AM}(s) + w_{LM} \log P_{LM} + \log W_{t,k}(s) + \lambda_{Pre} \log P_{pre}$$

where $\log P_{pre}$ is the output of the prefix module.

TABLE I
THE NUMBER OF DISEASE AND THE TERMINOLOGY IN TOTAL

Department	Number of Diseases	Terminology Number
Andriatria	20	18581
IM	45	35462
OAGD	29	17676
Oncology	42	20770
Surgical	89	23376

IM: Internal Medicine, OAGD: Obstetrics and Gynecology Department.

III. EXPERIMENTAL EVALUATION AND DISCUSSIONS

Experiments on a Chinese medical dialogue dataset[1] is tested in this section. Baseline methods (Transformer+beam search), Prefix Beam Search [26] and WFST Beam Search [27] are compared for performance evaluation.

A. Dataset and Experimental Setup

The dataset contains Chinese medical dialogue of 6 departments, including 792,099 questions and answers pairs (Q&A). For each department, the Q&A pairs are further divided into subcategories according to disease types, with 10∼89 main subcategories. The terminology set of each sentence is extracted using the TF-IDF [28] method. By means of the collected text data for sentence, a TF-IDF model is built and book-wise terminology with top n% of the words is extracted based on the scores. All stop words (single-letter and common phrases) are excluded from the list. Particularly, the number of extracted terminologies of 5 major departments is presented in Table I.

For performance evaluation, 225 sentences are recorded consisting of 200 minutes Chinese medical speech corpus. This speech corpuse is obtained from 15 students, each recording 15 sentences. The input acoustic features are 80-dimensional filter bank features. SpecAugment [29], [30] with mask parameters are selected. The convolution module [31] contains three 3×3 convolution blocks to downsample the input. The baseline Transformer used in this work has approximate 36.22M parameters. In training, the encoder and decoder have $N_e = 12$ and $N_e = 6$ layers, with 2048 units per layer. We set $d_{model} = 256$ and $H = 8$ for the multi-head attentions. Adam [32] with learning rate of 0.001, β_1 of 0.9, β_2 of 0.98 and ϵ of 1e-6, is used as the optimizer. The vocab size is 4233/5222 tokens in Aishell-1/Aishell-2. The Transformer backbones of [23] pre-trained on 178 hours of the Aishell-1 dataset and 1000 hours of the Aishell-2 dataset respectively are adopted, and then fine-tuned for 60 epochs with the label smoothing weight of 0.05 [33] and the schedule sampling weight of 0.1 [34]. The beam size of 5 is used in beam search with a length penalty of 0.1 [35]. To reduce the OOV problem, new words appearing in the Chinese medical dialogue medical dataset are added to the vocabulary and Chinese medical dialogue dataset are used to train LM.

B. Results and Comparisons

The results are reported in Tabel II where the first 5%, 10% and 15% of terms are boosted and the Transformer with beam searching is adopted as the baseline. As observed, using 15% of the terms for boosting is usually better than the other settings. Particularly, for 3 different boosting rates of the terminology (5%, 10%, 15%), the proposed model outperforms the baseline as well as the one only adopting SBPM, PM, respectively. Comparing with the baseline, the character error rates (CER) of Aishell-1 and Aishell-2 are enhanced from 22.90% and 12.71% to 18.73% and 9.36%, respectively, with the addition of 15% of terms.

TABLE II
CHARACTER ERROR RATES (CER) ON AISHELL-1 AND AISHELL-2 WITH LM WITH THREE DIFFERENT PERCENTAGES OF THE TERMINOLOGY IN BOOSTING

Models	w_{LM}	Pre Train Dataset					
		Aishell-1			Aishell-2		
		5%	10%	15%	5%	10%	15%
Baseline	0		22.90			12.71	
	1.0		20.77			11.13	
+SBPA	0	21.78	21.02	20.59	12.28	11.35	10.46
	0.5	20.68	19.93	19.76	11.33	10.24	9.97
	1.0	20.02	19.45	19.25	10.52	9.89	9.45
+PM	0	21.04	20.55	20.13	11.99	11.22	10.43
	0.5	20.23	20.01	19.84	11.02	10.12	9.78
	1.0	19.70	19.13	19.08	10.33	9.93	9.55
ours	0	20.63	20.28	19.56	11.33	10.45	10.02
	0.5	19.71	19.44	19.12	10.91	9.94	9.75
	1.0	**19.27**	**19.04**	**18.73**	**9.98**	**9.43**	**9.36**

Baseline method: Transformer+beam search.

TABLE III
THE RESULT WITH OTHER CONTEXTUAL BIASING WITH 10% TERMINOLOGY

Methods		Aishell-1		Aishell-2	
		w_{LM} = 0	w_{LM}=1.0	w_{LM}=0	w_{LM}=1.0
Baseline		22.90	20.77	12.71	11.13
Prefix Beam Search		**19.33**	**18.68**	12.03	10.05
WFST Beam Search		19.73	18.89	12.24	10.23
SBPA	PM				
✓		21.02	19.45	11.35	9.89
	✓	20.55	19.13	11.22	9.93
✓	✓	20.28	19.04	**10.45**	**9.43**

In addition to the ablation study, a comparison with several relevant ASR algorithms, including Prefix Beam Search [26], WFST Beam Search [27] and the baseline Transformer model are shown in Table III. Comparing with the baseline, the performance of three methods are all improved. Our model performs better than Prefix Beam Search and WFST Beam Search when using the Aishell-2 pretrain dataset, with the lowest overall CER 9.43%. When using Aishell-1 as the pretrain dataset, our method is slightly worse than Prefix Beam Search and WFST Beam Search.

IV. CONCLUSION

In this study, a speech transcription algorithm focuses on medical terminology was proposed. The proposed algorithm composed of ASR based on Transformer architecture, soft beam pruning algorithm (SBPA) and prefix word module (PM). The terminologies were weighted by the SBPA module and the PM module was then applied to avoid beam search from pruning the terminology at the beginning. Experiments on a Chinese medical dialogue dataset showed that our method was much more effective than LM in maintaining lower CER. The proposed ASR was helpful in areas with more terminology such as the telephone records and economic fields.

[1]https://github.com/Toyhom/Chinese-medical-dialogue-data, A real recognition demo is also submitted as an attachment of the letter.

REFERENCES

[1] D. Povey et al., "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Proc. Interspeech*, 2016, pp. 2751–2755.

[2] G. Hinton et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[3] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[4] D. Amodei et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 173–182.

[5] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 6645–6649.

[6] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2017, pp. 193–199.

[7] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[8] M. Sperber, J. Niehues, G. Neubig, S. Stüker, and A. Waibel, "Self-attentional acoustic models," in *Proc. Interspeech*, 2018, pp. 3723–3727, doi: 10.21437/Interspeech.2018-1910.

[9] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *Proc. IEEE 20th Conf. Oriental Chapter Int. Coordinating Committee Speech Databases Speech I/O Syst. Assessment*, 2017, pp. 1–5.

[10] J. Du, X. Na, X. Liu, and H. Bu, "Aishell-2: Transforming mandarin asr research into industrial scale," 2018, *arXiv:1808.10583*.

[11] E. Edwards et al., "Medical speech recognition: Reaching parity with humans," in *Proc. 19th Int. Conf. Speech Comput.*, 2017, pp. 512–524.

[12] C.-C. Chiu et al., "Speech recognition for medical conversations," 2017, in *Proc. Interspeech*, 2018, pp. 2972–2976, doi: 10.21437/Interspeech.2018-40.

[13] A. Mani, S. Palaskar, N. V. Meripo, S. Konam, and F. Metze, "ASR error correction and domain adaptation using machine translation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 16344–6348.

[14] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep context: End-to-end contextual speech recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2018, pp. 418–425.

[15] D. Zhao et al., "Shallow-fusion end-to-end contextual biasing," in *Proc, Interspeech*, 2019, pp. 1418–1422.

[16] N. Jung, G. Kim, and J. S. Chung, "Spell my name: Keyword boosted speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 6642–6646.

[17] M. Han et al., "Improving end-to-end contextual speech recognition with fine-grained contextual knowledge selection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 8532–8536.

[18] M. Jain, G. Keren, J. Mahadeokar, G. Zweig, F. Metze, and Y. Saraf, "Contextual RNN-T for open domain ASR," in *Proc. Interspeech*, 2020, pp. 11–15, doi: 10.21437/Interspeech.2020-2986.

[19] J. Tian, J. Yu, C. Weng, Y. Zou, and D. Yu, "Improving mandarin end-to-end speech recognition with word N-gram language model," *IEEE Signal Process. Lett.*, vol. 29, pp. 812–816, 2022.

[20] Y. M. Kang and Y. Zhou, "Fast and robust unsupervised contextual biasing for speech recognition," 2020, *arXiv:2005.01677*.

[21] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep context: End-to-end contextual speech recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2018, pp. 418–425.

[22] M. Han, L. Dong, S. Zhou, and B. Xu, "CIF-based collaborative decoding for end-to-end contextual speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 6528–6532.

[23] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 5884–5888.

[24] T. Nakatani, "Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration," in *Proc. Interspeech*, 2019, pp. 1408–1412.

[25] Z. Yao et al., "Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit," in *Proc. Interspeech*, 2021, pp. 4054–4058, doi: 10.21437/Interspeech.2021-1983.

[26] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2015, pp. 167–174.

[27] A. Y. Hannun, A. L. Maas, D. Jurafsky, and A. Y. Ng, "First-pass large vocabulary continuous speech recognition using Bi-directional recurrent DNNs," 2014, *arXiv:1408.2873*.

[28] K. Papineni, "Why inverse document frequency?," in *Proc. 2nd Meeting North Amer. Chapter Assoc. Comput. Linguistics*, 2001, pp. 1–8.

[29] D. S. Park et al., "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617, doi: 10.21437/Interspeech.2019-2680.

[30] D. S. Park et al., "Specaugment on large scale datasets," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 6879–6883.

[31] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3286–3295.

[32] D. P. Kingma and J. Ba, "Adam: A. method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[34] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1–9.

[35] Y. Wu et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, *arXiv:1609.08144*.