END-TO-END CONTEXTUAL SPEECH RECOGNITION USING CLASS LANGUAGE MODELS AND A TOKEN PASSING DECODER

Zhehuai Chen[*1,2], *Mahaveer Jain*[2], *Yongqiang Wang*[2], *Michael L. Seltzer*[2], *Christian Fuegen*[2]

[1]SpeechLab, Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Facebook, One Hacker Way, Menlo Park, CA 94025, USA
chenzhehuai@sjtu.edu.cn, {jainmahaveer,yqw,mikeseltzer,fuegen}@fb.com

ABSTRACT

End-to-end modeling (E2E) of automatic speech recognition (ASR) blends all the components of a traditional speech recognition system into a single, unified model. Although it simplifies the ASR systems, the unified model is hard to adapt when training and testing data mismatches. In this work, we focus on contextual speech recognition, which is particularly challenging for E2E models because contextual information is only available in inference time. To improve the performance in the presence of contextual information during training, we propose to use class-based language models (CLM) that can populate context-dependent information during inference. To enable this approach to scale to a large number of class members and minimize search errors, we propose a token passing algorithm with an efficient token recombination for E2E systems. We evaluate the proposed system on general and contextual ASR tasks, and achieve relative 62% Word Error Rate (WER) reduction for the contextual ASR task without hurting recognition performance for the general ASR task. We also show that the proposed method performs well without modification of the decoding hyper-parameters across tasks, making it a desirable solution for E2E ASR.

Index Terms— End-to-end Speech Recognition, Weighted Finite State Transducer, Token Passing, Class-based Language Model

1. INTRODUCTION

Automatic speech recognition (ASR) with Deep Neural Networks (DNN) commonly operates in a hybrid framework. There are a few models in this framework: DNNs, as discriminative acoustic models (AM), estimate the posterior probabilities of Hidden Markov Model (HMM) states; in the inference stage, external lexicons and language models (LM) are combined with AMs. All of these models are optimized independently [1]. *Weighted Finite State Transducer* (WFST) [2] has been proposed to combine different knowledge sources and perform search space optimization for efficient decoding, i.e., finding the sequence of labels that best matches the input audio. One way to perform efficient decoding is via *Token Passing*, which is a single-pass algorithm that can generate multiple alternatives for each WFST state [3].

Recently proposed E2E speech recognition has become popular as a result of both recent advances in neural modeling of context and history in sequences [4, 5], and more labeled training data for better generalization. In E2E speech recognition, a single model predicts words directly from input acoustics, which unifies the AMs, LMs, and lexicons into one system. Although E2E training benefits from sequence modeling and simplified inference [6, 7], it performs worse than traditional systems in long and noisy speech and needs a larger amount of transcribed acoustic data [8, 9] to perform well. Moreover, traditional language models and the corresponding decoding techniques are difficult to be incorporated into E2E systems [10, 11]. The inability to exploit knowledge from external language models and lexicons especially hampers the adaptability of E2E systems.

The ability to leverage external knowledge is particularly important for *Contextual Speech Recognition* [12] where contextual information can provide additional information about what a user may say and these information is only available at inference stage. Prior work in this field suffers from limited scalability in both context complexity and the amount of context phrases [13, 14], as discussed in Section 2. In this work we propose to model the context-specific information using a class-based LM (CLM) [15]. In this paradigm, contextual knowledge is modeled by an n-gram LM. Context phrases are composed on-the-fly into the CLM based WFST [16]. Because the WFST is non-deterministic, as discussed in Section 3.2, each E2E inference hypothesis includes alternative paths in the WFST. To handle these alternative paths, we propose a token passing decoder with an efficient token recombination for E2E systems for the first time.

The rest of the paper is organized as follows. In Section 2, prior works in E2E contextual speech recognition are briefly reviewed. Our main contributions are in Sections 3: i) use CLM to solve contextual speech recognition and wake word problem. ii) propose a token passing decoder for E2E inference. The relation to prior work is discussed in Section 4. Experimental results are presented in Section 5, followed by conclusion in Section 6.

2. CONTEXTUAL E2E SPEECH RECOGNITION

2.1. Attention-based End-to-end Modeling

We use the attention-based encoder-decoder model [17] for E2E modeling. It predicts the posterior probability of label sequences given both a feature sequence $\mathbf{x}$ and previous inference labels $\mathbf{l}_{1:i-1}$.

$$P(\mathbf{l}|\mathbf{x}) = \prod_i P(l_i|\mathbf{x}, \mathbf{l}_{1:i-1}) \tag{1}$$

$$\mathbf{h} = \text{Encoder}(\mathbf{x}) \tag{2}$$

$$P(l_i|\mathbf{x}, \mathbf{l}_{1:i-1}) = \text{AttentionDecoder}(\mathbf{h}, \boldsymbol{s}_{i-1}) \tag{3}$$

where $\mathbf{h}$ is the sequence of encoder states and $\mathbf{s_{i-1}}$ is the decoder hidden state from the previous time step. The Encoder$(\cdot)$ is typically a unidirectional or bidirectional long short term memory (LSTM) network while the AttentionDecoder$(\cdot)$ is a unidirectional LSTM.

Compared to traditional decoder in the hybrid system [1], the AttentionDecoder$(\cdot)$ implicitly captures LM information in a way that is jointly trained with the Encoder which can be interpreted as

[*] This work was done when the first author was an intern with Facebook.

an acoustic model. Because of this tight unification between models and decoder, such E2E systems cannot be easily adapted to new domains or contexts. In contrast, traditional systems can do this easily via updates to the language model [12].

2.2. On-the-fly Rescoring with External WFST

A contextual automatic speech recognition (ASR) system dynamically incorporates real-time context into the recognition process of a speech recognition system [12]. A typical example of contextual information is the personal information such as a user's contacts.

One branch of methods is to generate an on-the-fly contextual LM and include it into the recognition process to bias the beam search in Section 2.1. [13] introduces the shallow fusion approach.

$$\mathbf{l}^* = \arg\max_{\mathbf{l}} \log P(\mathbf{l}|\mathbf{x}) + \lambda \log P_C(\mathbf{l}) \tag{4}$$

where $P_C(\mathbf{l})$ is the introduced contextual LM and λ is a scaling factor. [13] proposes to use similar on-the-fly rescoring technique as [16] to obtain $P_C(\mathbf{l})$. The method shows good performance in limited number of context phrases but the recognition accuracy starts to drop when the number of contexual phrases is above 100. One possible reason could be that it follows the WFST search idea from [16] to traverse epsilon arcs only in the absence of a matching symbol. This can introduce significant search errors inside the word class [1]. We will look into this problem and propose methods to alleviate it in Section 3.2.

2.3. Contextual E2E Modeling

Another method that try to integrate contextual information into the E2E modeling is called CLAS [14]. This technique first embeds each phrase, represented as a sequence of graphemes, into a fixed-dimensional representation. And then it employs an attention mechanism to summarize the available context at each step of the output predictions. By this way, CLAS explicitly models the probability of seeing particular phrases given audio and previous labels.

To scale up this paradigm and make it into use, two fundamental problems need to be considered: i) Model the similarities between large amounts of context phrases. Although [14] proposes a conditioning mechanism to reduce the amount of phrases considered, a better and more unified solution is important to the scaling up [14]. ii) Constrain the search space of context phrases at a particular step in AttentionDecoder$(\cdot)$. The above attention mechanism is done by using all phrases, while general CLM [15] applied in this paper, only uses contextual phrases that are relevant at current prediction step.

3. THE PROPOSED METHOD

3.1. Class-based Language Model and WFST

This work follows the paradigm and formulation of the shallow fusion in Section 2.2. To solve the extendibility in modeling complex context, we first extend the paradigm by using CLM [15]. CLM refers to introducing word equivalence classes into n-gram LM. In contextual speech recognition, the contextual phrases, e.g. a user's favorite songs and contacts, can be grouped into multiple word equivalence classes (*inside the class*). And the context of the conversation is modeled by n-gram LM (*outside the class*).

We compile n-gram contexts with word equivalence classes(call @name) and, contextual phrases(Tom Cruise, Lady Gaga) of each word equivalence class in separate WFST graphs. These WFST graphs are then composed with the "speller" WFSTs [13] to obtain the grapheme level WFSTs. We do determinization operation on WFSTs of contextual phrases to reduce number of tokens as discussed in the next section. In the inference stage, a form of on-the-fly composition between the *inside* and *outside the class* WFSTs is conducted [16], without requiring any changes to the pre-compiled transducers [18]. Figure 1 shows an example.

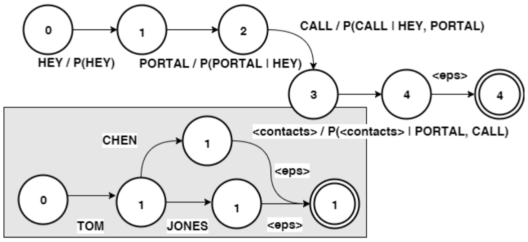


Fig. 1. *Examples of CLM in Contextual Speech Recognition (word level WFSTs for simplicity). "HEY / P(HEY)" denotes the symbol of arc is "HEY" with the probability P(HEY). States in the shaded box represent names in the word class "$\langle contact \rangle$".*

Our proposed framework can also improve the wake word recognition in the E2E system. We add a special word class in the start of the sentence with a boosting factor (*keyword boosting*; tuned on the development set). In the inference stage, the wake word grapheme sequence is composed into word equivalence class similar to context phrases.

3.2. Token Passing Decoder

In [16], it assumes that weight of matched-symbol arcs would always be lower than backoff arcs. Because of this assumption, n-gram LM WFST can be treated as deterministic and a single token decoder can be used.

Grapheme level WFST is non-deterministic because of following reasons: i) n-gram LM has backoff transitions. ii) Duplication of phrases between two word equivalnce classes. iii) Duplication of phrases between a word equivalnce class and n-gram LM words. ii) and iii) are issues only because of on-the-fly composition as we never determinize the whole WFST. Figure 2(a) shows an example. Because of non-deterministic WFST, each graphemes hypothesis of E2E inference can have multiple paths in WFST. To cope with multiple tokens for a hypothesis, we proposed a token passing decoder for E2E system in Algorithm 1, with efficient token recombination. Figure 2 shows examples of how to process tokens in our algorithm.

In algorithm 1, the k-th token in the token set $\mathcal{H}$ is defined as a 4-element tuple $(\boldsymbol{s}_k, \mathbf{l}_k, t_k, q_k)$, whereas $\boldsymbol{s}_k$ is the k-th decoder hidden state $\boldsymbol{s}_{i-1}^{(k)}$ at the last prediction step in Equation (3); $\mathbf{l}_k$ is the partially-decoded sequences; t_k is the last state of WFST path whose output sequence is $\mathbf{l}_k$. Note that as discussed above, there could be multiple t_k associated with the same $\mathbf{l}$; q_k is the score for the k-th partial hypothesis. At each decoding step, we expand k-th token by concatenating $\mathbf{l}_k$ with every grapheme: $\boldsymbol{s}_k$ and q_k are first updated in Line 9 and 10. WFST states are then expanded in Line 12 to 15: $\mathrm{SearchFST}(t_k, l)$ returns all the possible WFST states (and the correspoding cost) which can be reached by departing from the state t_k and consuming the input symbol l; again multiple tokens can exist with the new hypothesis output $\mathbf{l}'$. $\mathrm{TokenRecombination}$ function is proposed in Line 16: for every $(\mathbf{l}'_k, t'_k)$ pair, it is only necessary to maintain the best token; for the same partial hypothesis $\mathbf{l}'_k$, we only maintain $B_{\texttt{tok}}$ tokens at most. Finally, after expanding all the current tokens in $\mathcal{H}$, we select the best B partial hypotheses $\mathbf{l}'_k$ in SelectTopN function. For the same partial hypothesis, we need to maintain multiple tokens with different WFST states in our proposed algorithm but its time complexity is similar to standard beam search

[1] [16] introduces a special backoff method on word level. Beside that, [13] does not describe any further design on its "speller" WFST inside the word.

as discussed later. Compared to the on-the-fly rescoring [13], an important difference of our algorithm is:

- Multiple tokens. In the WFST search (line 12), previously proposed on-the-fly rescoring [16] traverses epsilon transitions only in the absence of a matching symbol. Figure 2(a) shows an example where this can introduce search errors. For the grapheme ”C”, our proposed method would have two tokens corresponding to state 5 and 6. State 6 extends from the backoff state 0, which can be traversed from state 4. In [13], state 0 is not traversed because state 4 already has a matched arc to state 5. This results in not exploring state 6 and hence introducing a possible search error. In this work, we propose to keep multiple tokens for each hypothesis from E2E inference using token passing method.

Algorithm 1: Token Passing Algorithm for E2E Model

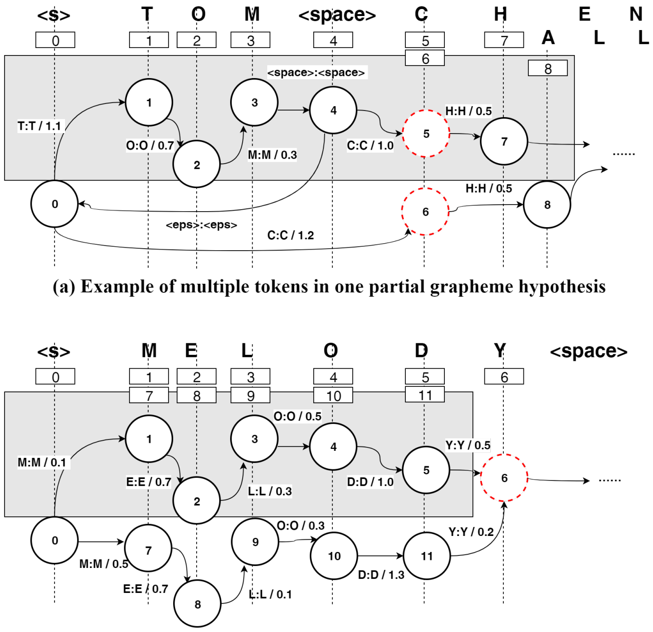
```
1  Input: $\mathbf{h}$, defined in Equation (2)
2  Initialization: $\mathcal{H} = \{(s_0, \texttt{"<bos>"}, \texttt{FST.start}, 0)\}$ ;
3  while EndDetection($\mathcal{H}$) [19] [2] do
4      $\mathcal{H}' \leftarrow \{\}$;
5      for $(s_k, \mathbf{l}_k, t_k, q_k) \in \mathcal{H}$ do
6          for each grapheme $l$ do
7              $\mathcal{H}_l = \{\}$ ;
8              • extend decoder network by grapheme l
9                  $\mathbf{l}'_k \leftarrow \mathbf{l}_k + l$; $q'_k \leftarrow q_k + p(l|s_k, \mathbf{h})$;
10                 $s'_k \leftarrow$ UpdateDecoderState$(s_k, l)$ ;
11             • extend FST state
12                 $\mathcal{T}'_k \leftarrow$ SearchFST$(t_k, l)$;
13                 for $(t'_k, p'_k) \in \mathcal{T}'_k$ do
14                     $\mathcal{H}_l \leftarrow \mathcal{H}_l \cup \{(s'_k, \mathbf{l}'_k, t'_k, q'_k + \lambda p'_k)\}$;
15                 end
16             $\mathcal{H}_l \leftarrow$ TokenRecombination$(\mathcal{H}_l, B_{\texttt{tok}})$;
17             $\mathcal{H}' \leftarrow \mathcal{H}' \cup \mathcal{H}_l$
18         end
19     end
20     $\mathcal{H} \leftarrow$ SelectTopN$(\mathcal{H}', B)$
21 end
22 return best path in $\mathcal{H}$;
```

- Token Recombination. To reduce the amount of tokens in each hypothesis and add diversity in WFST paths, token recombination is proposed for the decoder. Tokens can be combined only if they have both the same WFST state and E2E state. As different hypotheses have different E2E states, the token recombination can only be conducted on tokens of one hypothesis, e.g. Figure 2(b).

The proposed method does not change the computational complexity of standard beam search in E2E inference. The complexity of standard E2E inference is $C_{E2E} = O(B \cdot L \cdot D) + O(F)$, where B is the beam size of hypotheses , L is the length of the sequence, D is the complexity of the decoder neural network of E2E and, F is the complexity of encoder neural network of E2E. The token passing decoding does not change the beam search in E2E inference. Complexity of token passing decoding is $C_{DEC} = O(B \cdot L \cdot B_{tok} \cdot U)$, where B_{tok} is the beam of WFST tokens in each hypothesis, U is the size of grapheme. Since $D \gg B_{tok} \cdot U$, we have $C_{DEC} + C_{E2E} \approx C_{E2E}$.

4. RELATION TO PRIOR WORK

In the inference stage of the E2E speech recognition, prior work such as [20, 21, 22, 23, 10, 24] uses n-gram LM or NNLM to bias search space. In contextual E2E speech recognition, because of the mismatch between training and test utterances, it is even more important to integrate external knowledge sources to improve the WER. [13] proposes to use on-the-fly composed external contextual LM to bias the beam search of E2E inference. Another branch of methods [14] tries to model the probability of seeing particular context phrase given audio and previous labels. The prior work in this field suffer from limited extendibility in both context complexity and the amount of context phrases. The advantages of this work include: i) Better generalization of complex context by using CLM. ii) Less search errors by keeping E2E and WFST states separately using multiple tokens with recombination in 1-pass decoding.



(a) Example of multiple tokens in one partial grapheme hypothesis

(b) Example of token recombination in one partial grapheme hypothesis

Fig. 2. *Examples of the Token Passing Algorithm. States in the shaded box are of the word class and they are generated from on-the-fly composition. The small boxes below graphemes denote their tokens and state numbers. In (a), the sixth grapheme ”C” is the prefix of ”CHEN” and ”CALL”, corresponded to two tokens in different WFST states (red dash circles). In (b), there are two ”MELODY” in both* inside the class *and* outside the class*. Their tokens can be recombined in state 6, where they are with the same history states in both E2E and WFST.*

5. EXPERIMENTS

5.1. Setup

The data is collected with the help of crowd sourced workers. These workers were asked to write and speak utterances that they could ask an AI assistant. Each utterance could belong to general speech [3] or one of many possible domains [4]. We have 10 million utterances for training. Size of *General* ASR testset is 60 thousand whereas size of *Contextual* ASR testset is 10 thousand. In order to generate possible members of CLM for *Contextual* ASR testset, we first extract the true entity from the utterance and then add 999 fake entities of the same type. Each utterance of *Contextual* ASR testset has wakeup word in the beginning.

In training, 40-dimensional fiterbank feature is extracted with a

[2] We also force hypotheses to end in the end of WFST.

[3] e.g. what are the ingredient in pork stew?

[4] e.g. play Lady Gaga song (music domain), call Alex (calling domain).

frame rate of 10ms. E2E models with grapheme units were built with PyTorch [25] based on Espnet [26]. 2-layer BLSTM with 1400 nodes is used for the encoder, and 2-layer LSTM with 700 nodes is used for the decoder. The model is optimized by both connectionist temporal classification (CTC) and E2E criteria [19]. Our n-gram LM is 3-gram LM trained on vocabulary of 300 thousand words. We use value of 10 for both B (hypothesis beam size) and B_{tok} (WFST token beam size). Word error rate (WER) is used as the metric for evaluation.

5.2. Performance Comparison

We compare performance of the E2E system for both *Contextual* and *General* ASR testsets. Decoding hyper-parameters used for both testsets are same in each of the row in Table 1. First row shows experiments without using any LM for E2E system. The WER for *General* ASR testset is 5.9 whereas WER for *Contextual* ASR test is 34. The hardness of contextual speech recognition stems from: i) Lack of wake word modeling in training set. This affects both the recognition of wake word and history modeling [5] for the remaining words. ii) Lack of contextual phrases in training utterances.

Table 1. *WER of End-to-end ASR in General and Contextual ASR.*

system	General	Contextual
E2E	5.9	35.1
+ n-gram LM	**5.6**	31.4
+ Class LM	5.7	**13.5**

The second row shows experiments with an external 3-gram LM for E2E system. Decoding is performed by the proposed token passing decoder. For the *General* ASR task, WER improves to 5.6% from 5.9%. For *Contextual* ASR task, WER improves from 35.1% to 31.4%. The improvement of general ASR from external n-gram LM is consistent with the results in [23]. We do not examine NNLM as the main purpose of this work is to improve the contextual ASR. Improvement for *Contextual* ASR tasks results from boosting wake word as discussed in the end of Section 3.1. Nevertheless, simply boosting the scores of wake word can only help the recognition of wake word, it does not solve the problem of history state mismatch of the remaining words. Traditional LSTM-HMM systems trained with a cross-entropy criterion gets 5.6% WER on *General* ASR testset for same n-gram LM.

n-gram LM can easily be integrated with the CLM based paradigm as discussed in Section 3.1. Experiemtns for the proposed CLM based token passing decoder is in the third row. It achieves similar performance [6] as n-gram LM for *General* ASR testset but achieves significant improvement for *Contextual* ASR testset(from 31.4% to 13.5% WER). These improvements comes from 1) modelling context using CLM and 2) reducing search errors by token pass decoder. We conduct more analysis in the next section. CLM is also good at adaptability as shown by comparable results with n-gram LM for *General* ASR testset.

5.3. Analysis

We firstly show the effectiveness of the proposed method compared to the shallow fusion [13] based systems. Previous shallow fusion based systems essentially has value of B_{tok}(beam size of WFST tokens) as 1. Figure 3 shows relationship between WER and B_{tok}. With B_{tok} less than 5, the system has significantly worse performance [7], which is consistent with the performance degradation with more than 1K phrases in [14]. This shows the importance of using multiple tokens in the WFST beam.

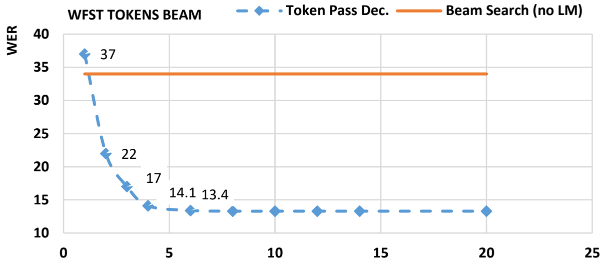


Fig. 3. *WER v.s. Beam of WFST Tokens in the Token Passing Decoder.*

In Figure 4, we show experiments for scaling up number of contextual phrases. Though increasing the number of contextual phrases degrades the WER, we still have acceptable WER for upto 5K phrases, which is acceptable for most of the real world applications. After around 8K phrases, the system breaks down.

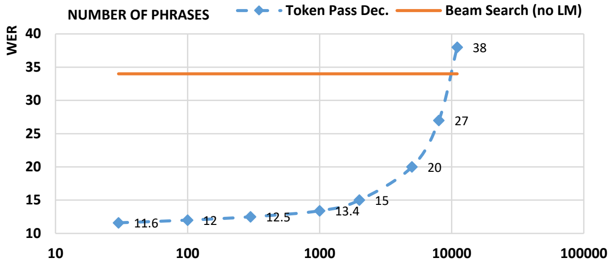


Fig. 4. *WER v.s. Number of Phrases (beam of WFST tokens is 10).*

Finally, we show impact on WER for *General* ASR when we tune hyper-parameter to improve *Contextual* ASR. The curves in Figure 5 are mostly smooth, which shows general ASR performance is not sensitive to the contextual ASR performance and, vice versa.

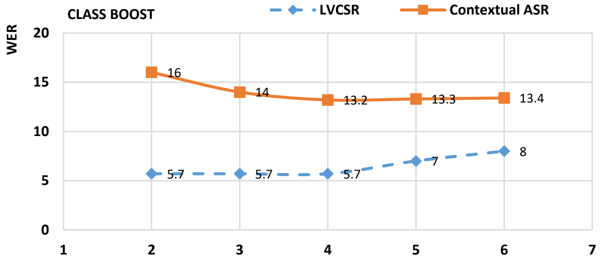


Fig. 5. *class boosting*, defined in Section 3.1 and its Effects in General and Contextual ASR. Similar trend in *keyword boosting*.

6. CONCLUSION

In this work, we propose to (a) use CLM to solve contextual speech recognition with (b) token passing decoder for E2E inference. The result on contextual ASR achieves consistent and significant improvements. Future works include extendibility to large number of context phrases and combining NNLM [27, 28].

[5] If the model does not see the wake word during training, it cannot recognize both the wake word and speech after the wake word while decoding because of the unseen history state

[6] The slight difference stems from more WFST branchings of contextual phrases in CLM based WFST.

[7] Notably, we do not observe this phenomenon in general ASR [13]. We believe the different observation can be the over-biasing stems from the perplexity difference between word classes in CLM.

7. REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[3] S. J. Young, N. H. Russell, and J. H. S. Thornton, *Token passing: a simple conceptual model for connected speech recognition systems*, Cambridge University Engineering Department Cambridge, UK, 1989.

[4] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[5] Z.-H. Chen, J. Droppo, J. Li, and W. Xiong, "Progressive joint modeling in unsupervised single-channel overlapped speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 1, pp. 184–196, 2018.

[6] Z.-H. Chen, Y. Zhuang, Y. Qian, and K. Yu, "Phone Synchronous Speech Recognition With CTC Lattices," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 86–97, Jan 2017.

[7] Z.-H. Chen, Y. Qian, and K. Yu, "A unified confidence measure framework using auxiliary normalization graph," in *International Conference on Intelligent Science and Big Data Engineering*. Springer, 2017, pp. 123–133.

[8] Hagen Soltau, Hank Liao, and Hasim Sak, "Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016.

[9] Mingkun Huang, Yongbin You, Zhehuai Chen, Yanmin Qian, and Kai Yu, "Knowledge distillation for sequence model.," in *Interspeech 2018*, 2018.

[10] Takaaki Hori, Shinji Watanabe, and John R Hershey, "Multi-level language modeling and decoding for open vocabulary end-to-end speech recognition," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 287–293.

[11] Zhehuai Chen, Qi Liu, Hao Li, and Kai Yu, "On modular training of neural acoustics-to-word model for lvcsr," in *IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, Calgary, Canada, April 2018.

[12] Ian McGraw, Rohit Prabhavalkar, Raziel Alvarez, Montse Gonzalez Arenas, Kanishka Rao, David Rybach, Ouais Alsharif, Haşim Sak, Alexander Gruenstein, Françoise Beaufays, et al., "Personalized speech recognition on mobile devices," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5955–5959.

[13] Ian Williams, Anjuli Kannan, Petar Aleksic, David Rybach, and Tara N Sainath, "Contextual speech recognition in end-to-end neural network systems using beam search," *Proc. Interspeech 2018*, pp. 2227–2231, 2018.

[14] Golan Pundak, Tara N Sainath, Rohit Prabhavalkar, Anjuli Kannan, and Ding Zhao, "Deep context: end-to-end contextual speech recognition," *arXiv preprint arXiv:1808.02480*, 2018.

[15] Reinhard Kneser and Hermann Ney, "Improved clustering techniques for class-based statistical language modelling," in *Third European Conference on Speech Communication and Technology*, 1993.

[16] Keith Hall, Eunjoon Cho, Cyril Allauzen, Francoise Beaufays, Noah Coccaro, Kaisuke Nakajima, Michael Riley, Brian Roark, David Rybach, and Linda Zhang, "Composition-based on-the-fly rescoring for salient n-gram biasing," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[17] William Chan, *End-to-End Speech Recognition Models*, Ph.D. thesis, Carnegie Mellon University Pittsburgh, PA, 2016.

[18] Paul R Dixon, Chiori Hori, and Hideki Kashioka, "A specialized wfst approach for class models and dynamic vocabulary," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[19] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[20] Z.-H. Chen, W. Deng, T. Xu, and K. Yu, "Phone Synchronous Decoding with CTC Lattice," in *Interspeech 2016*, 2016, pp. 1923–1927.

[21] Zhehuai Chen, "Linguistic search optimization for deep learning based lvcsr," *arXiv preprint arXiv:1808.00687*, 2018.

[22] Zhehuai Chen, Justin Luitjens, Hainan Xu, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur, "A gpu-based wfst decoder with exact lattice generation," *arXiv preprint arXiv:1804.03243*, 2018.

[23] Anjuli Kannan, Yonghui Wu, Patrick Nguyen, Tara N Sainath, Zhifeng Chen, and Rohit Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," *arXiv preprint arXiv:1712.01996*, 2017.

[24] Yue Wu, Tianxing He, Zhehuai Chen, Yanmin Qian, and Kai Yu, "Multi-view lstm language model with word-synchronized auxiliary feature for lvcsr," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pp. 398–410. Springer, 2017.

[25] Adam Paszke, Sam Gross, and Soumith Chintala, "Pytorch," 2017.

[26] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[27] Ke Li, Hainan Xu, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur, "Recurrent neural network language model adaptation for conversational speech recognition," *Proc. Interspeech 2018*, pp. 3373–3377, 2018.

[28] Da Zheng, Zhehuai Chen, Yue Wu, and Kai Yu, "Directed automatic speech transcription error correction using bidirectional lstm," in *Chinese Spoken Language Processing (ISCSLP), 2016 10th International Symposium on*. IEEE, 2016, pp. 1–5.