INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Class LM and Word Mapping for Contextual Biasing in End-to-End ASR

Rongqing Huang, Ossama Abdel-hamid, Xinwei Li, Gunnar Evermann

Apple

{huangr, oabdelhamid, xinwei_li2, gevermann}@apple.com

Abstract

In recent years, all-neural, end-to-end (E2E) ASR systems gained rapid interest in the speech recognition community. They convert speech input to text units in a single trainable Neural Network model. In ASR, many utterances contain rich named entities. Such named entities may be user or location specific and they are not seen during training. A single model makes it inflexible to utilize dynamic contextual information during inference. In this paper, we propose to train a context aware E2E model and allow the beam search to traverse into the context FST during inference. We also propose a simple method to adjust the cost discrepancy between the context FST and the base model. This algorithm is able to reduce the named entity utterance WER by 57% with little accuracy degradation on regular utterances. Although an E2E model does not need a pronunciation dictionary, it's interesting to make use of existing pronunciation knowledge to improve accuracy. In this paper, we propose an algorithm to map the rare entity words to common words via pronunciation and treat the mapped words as an alternative form to the original word during recognition. This algorithm further reduces the WER on the named entity utterances by another 31%.

Index Terms: End-to-End Speech Recognition, Contextual Biasing, Word Mapping Through Pronunciation

1. Introduction

Conventional speech recognition systems include several main components: acoustic model, language model, and pronunciation dictionary. Each of them is separately constructed and optimized. In recent years, an all-neural, end-to-end (E2E) model that directly converts speech into text through a sequence model became popular. Instead of separately optimized components, the E2E model is a single trainable neural network. It removes the HMM assumptions and enables end-to-end optimization. Architectures like Connectionist Temporal Classification (CTC) [1], attention based sequence models such as Listen, Attend and Spell (LAS) [2], Recurrent Neural Network Transducer (RNN-T) [3] have obtained impressive results. In particular, architectures like LAS can sometimes outperform the conventional system [4, 5]. The base LAS model requires the whole input sequence before it can compute the attention. This makes it infeasible for online streaming application. In [6, 7], a Monotonic Chunkwise Attention (MOCHA) was proposed for the attention based sequence-to-sequence model. The LAS-MOCHA structure becomes the base model for our study, although our proposed methods should be directly applicable to other types of E2E model.

Users' voice requests often involve personal content like contact names, app names, music titles, etc. There are a few issues here. First, such personal content frequently includes rare and foreign words. The general training set has very few occurrences of such words. Second, each user's personal content is different. The common entities may not be what the users want. For example, if a user has a contact `Jain Smith`, a phrase like `call Jain Smith` may be misrecognized as `call Jane Smith`. Another user may have a contact `Jaine Smith`. It's important to inject the user-specific content during inference.

In conventional hybrid HMM-DNN system, the contextual information is usually represented as a Weighted Finite-State Transducer (WFST, [8]), and injected into the main FST graph during the recognition [9, 10, 11].

A single E2E model lacks the flexibility to inject the contextual information during recognition. There have been various attempts to improve it, e.g. in [12], a separate attention component is added to model the contextual phrases (a.k.a. bias phrases); in [13, 14, 15], an on-the-fly rescoring (a.k.a. shallow fusion) with bias phrases was proposed. Our proposed method is similar to [13, 15] with a few important differences. We identify the bias phrases in the training data and model the transitions between regular words and bias words, while during inference, the user-specific bias phrases are inserted as WFST graphs at the relevant place in the beam search. We propose to normalize the scores between the paths from the contextual bias FST and from the base search space. It's hard for the E2E system to output words it has never seen in the training data [16, 17]. Such words are especially prevalent in user contact lists and music lists. We propose a method to do word mapping, i.e, transform the rare words to common words through pronunciation. This method makes use of an existing lexicon and achieves nice accuracy gain.

2. Base LAS-MOCHA Model

Given the speech sequence $\mathbf{x} = \{x_1, ..., x_L\}$ with length L, and output word sequence $\mathbf{y} = \{y_1, ..., y_U\}$ with length U, the E2E model computes the probability

$$P(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{U} P(y_i|\mathbf{x}, y_1, y_2, ..., y_{i-1}) \quad (1)$$

An encoder converts $\mathbf{x}$ to intermediate outputs $\mathbf{h} = \{h_1, ..., h_T\}$ through a Recurrent Neural Network (RNN), typically an LSTM. One important aspect for ASR is there are many more speech frames than the number of output tokens, usually there is a reduction factor N, thus $T = \frac{L}{N}$.

$$h_j = \text{EncoderRNN}(x_j, h_{j-1}) \quad (2)$$

The decoder is also an RNN. It takes encoder outputs $\mathbf{h}$ (a.k.a. memory), and previous output token y_{i-1}, generates the current decoder state s_i:

$$s_i = \text{DecoderRNN}(y_{i-1}, s_{i-1}, c_i) \quad (3)$$

which is then passed through a generation network, typically a feedforward network with softmax output, to produce the next

http://dx.doi.org/10.21437/Interspeech.2020-1787

output token y_i:

$$y_i = \text{Generate}(s_i, c_i) \tag{4}$$

c_i is the vector at decoder step i that summarizes information from the encoder:

$$c_i = \sum_{j=1}^{T} \beta_{i,j} h_j \tag{5}$$

Where $\beta_{i,j}$ is attention weight at output step i on j-th encoder output. The difference between original LAS and the MOCHA attention is how $\beta_{i,j}$ is computed. MOCHA defines two levels of score functions for attention weight computation. Please refer to Section 2 in [7] for details.

3. Context Injection

3.1. Context Aware Training

Our LAS-MOCHA model takes 40-dimension mel-filter banks, and outputs BPE tokens (Byte Pair Encoding [18]). With BPE tokens, we can in theory cover all the words in a language, so we don't have out-of-vocabulary (OOV) problems. This is important for the user-specific named entities. From the speech-text pair training data, we relabel the transcription to insert class LM tags. E.g, for utterance `call Jain Smith mobile`, it's relabelled into `call @contact# Jain Smith #contact@ mobile`; similarly for utterance `open ClassDojo`, it's relabelled into `open @app# ClassDojo #app@`. The tokens `@contact#`, `@app#`, `#contact@`, `#app@` are class enter and exit tokens for classes `contact` and `app` respectively. The class tags are excluded from BPE processing. Human transcription is based on existing speech recognition system output that includes class LM tags. We apply an edit distance alignment between transcription and recognition output, then insert the class LM tags into the transcription. Alternatively, a process like named entity tagging can be used. Note that tagging does not need to be perfect as the content between the enter and exit tags will be replaced by user-specific FST during inference.

3.2. Contextual Bias FST Construction

The contextual bias FST is a transducer from BPE subword sequence to word sequence (phrase). The cost on the FST arcs is derived from the relevant frequency f_i for phrase i:

$$C_i = -\log \frac{f_i}{\sum_{j=1}^{N_l} f_j} \tag{6}$$

where N_l is the number of phrases in this context FST. If there are M arcs on the path for this phrase, actual cost on each arc is

$$C_i' = \frac{C_i}{M} \tag{7}$$

The transducer T is determinized and minimized:

$$T_c = \text{Min}(\text{Det}(T)) \tag{8}$$

We build one FST per class. Each use case (user-specific, location-specific, etc.) will have its own FSTs.

3.3. Inference

The contextual bias FST is constructed before inference. During inference, if an active token in the beam is one of the class enter tags like `@contact#`, the corresponding context FST will be activated and its path will be traversed. Such path will compete against other paths inside the context FST, and also at the same time, compete against the paths in the base search space. When a final state of a context FST is reached, it traverses the class exit tag `#contact@`, and gets back to the base search space. For t-th step of beam search, the score S_i^t for i-th beam is:

$$S_i^t = \log P_b(y_{i,t}|y_{i,<t}, \mathbf{x}) + \lambda_c \log P_c(y_{i,t}|y_{i,<t}) \tag{9}$$

where $P_b(\cdot)$ is probability from the base E2E model and $P_c(\cdot)$ is probability from the contextual bias FST, and λ_c is the scale on the context FST. For the path outside of the context FST, the second term is 0, this makes the context FST path always worse than the base path (since log of probability is a negative number). Therefore, we define the following scoring function for paths outside of the context FST:

$$S_i^t = \log P_b(y_{i,t}|y_{i,<t}, \mathbf{x}) + \lambda_b \Gamma_t \tag{10}$$

Where λ_b is a scale, and Γ_t is the normalization score at step t:

$$\Gamma_t = \frac{1}{\kappa_t} \sum_{i=1}^{\kappa_t} \log P_c(y_{i,t}|y_{i,<t}) \tag{11}$$

Where κ_t is the number of active paths from the context FSTs at step t. If all the paths at step t are in the base search space, $\Gamma_t = 0$; if any paths at step t are inside a context FST, Eq. 9 defines S_i^t for paths inside the context FST, and Eq. 10 defines S_i^t for paths in the base search space. Note that the actual ranking of the paths at step t are based on the accumulated score:

$$S_i^{1,\ldots,t} = \sum_{k=1}^{t} S_i^k \tag{12}$$

4. Word Mapping Through Pronunciation

The E2E system can output graphemes or words, therefore it does not need a pronunciation dictionary. On the other hand, we have created a large pronunciation dictionary in the many years of conventional system development. The grapheme-to-phoneme (G2P) system is also mature. It is an interesting research question how to inject such knowledge into the E2E system. Rare or even foreign words are common in the named entity phrases. The E2E system has difficulty generating words it rarely sees. In this section, we describe a method to convert the rare named entity words into more common words through pronunciation so they are easier to recognize.

Given a word n-gram model G trained from the text data used in the E2E model training, and a pronunciation dictionary (lexicon) L, we construct

$$D = L \circ G \tag{13}$$

For a word W, its pronunciation is represented as a phoneme FST P_W, the pronunciation is either from a human generated lexicon or a G2P system and it may have multiple pronunciations for W (the rare word). A new word W' (the common word) is obtained through

$$W' = \text{TopSort}(\text{ShortestPath}(P_W \circ D)) \tag{14}$$

Then the subword list corresponding to W' is used as the subword list for W for contextual bias FST construction. We will also show experiments that adding the list to W instead of replacing the original one.

5. Experiments

We use English data in all our experiments. The base E2E model is LAS-MOCHA. The encoder is 5-layer unidirectional LSTM, 1400 cells, with 700-dim projections. The 40-dim mel-filter bank is the input to the encoder. 4-head, 800-dim, chunk-size 2 MOCHA attention is used. The decoder uses 2-layer unidirectional LSTM, each has 800 cells, and 400-dim projection. The text is processed with 6.4K BPE tokens. We use two test sets, first is a named entity rich set (*named entity set*) with 13K utterances. Sample utterances include "`call Jain Smith mobile`" and "`open ClassDojo`"; the other has no named entities (*regular set*) and has 21K utterances. Note that the user-specific context FSTs are loaded even for the *regular set* so this is a good test set for measuring the "anti-biasing" phenomenon. Both test sets are typical voice assistant requests.

The E2E model is trained with block-momentum SGD [19], 32 V100 GPUs, 20000 frames per minibatch, L2 normalization on gradient, and cross-entropy criterion, with label smoothing [20] weight 0.05 and schedule sampling [21] weight 0.1. The initial learning rate is 0.025 and decayed by factor 0.8 when no improvement observed on the validation set. The model was trained for 25 epochs. We use beam size 8 during beam search, and length penalty [22] 0.1.

5.1. Contextual bias FST in inference

The first set of experiments are to confirm the effectiveness of the proposed context FST algorithm in the inference. The amount of training data for the E2E model is a few thousand hours here.

Table 1: *Effect of adding class LM tags.*

Setup	Named entity set WER	Regular set WER
LAS-MOCHA	19.2	9.1
+class LM tags	18.1	8.4

The class LM enter-exit tags can help the model learn the transition between regular words and named entity words. This is evident from the results in Table 1, where only the class LM tags were added to the training text and the accuracy is already improved for both named entity test set and regular test set.

Next, the contextual bias FST is added in the inference. When $\lambda_c = 0$, it means no context FST is used; when $\lambda_b = 0$, it means no score normalization is used. Table 2 shows the impact of using context FSTs and score normalization. From setup 1 to 2, using context FST and scale 0.1 reduces the WER from 18.1 to 14.4; from setup 2 to 3, turning on score normalization alone reduce the WER from 14.4 to 11, changing the context FST scale from 1.0 to 0.1 further reduces the WER to 9.0. The overall WER reduction of using context FST is 50% (from 18.1 to 9.0). The last column shows the WER on the *regular set*, the WER is increased slightly from 8.4 to 8.5.

5.2. Word mapping through pronunciation

The E2E model used in this experiment is trained with four times more data, SpecAugment [23], and Minimum Word Error Rate criterion (MWER, [24, 25]), the weight on the cross-entropy loss during MWER is 0.05. We ran one epoch of MWER after the cross-entropy training. The model parameters and other hyper-parameters are the same as in Section 5.1. For word mapping, the unigram trained on the text portion of the training data is used and the base lexicon has less than a million unique words. A bi-LSTM based G2P was used to generate pronunciations for the OOV words in the user's context lists. The results are shown in Table 3.

Table 2: *Score normalization of contextual bias FST and its impact on accuracy.*

ID	λ_c	λ_b	Named entity set WER	Regular set WER
1	0	0	18.1	8.4
2	0.1	0	14.4	8.5
3	1.0	1.0	11.0	8.5
4	0.1	1.0	9.0	8.5

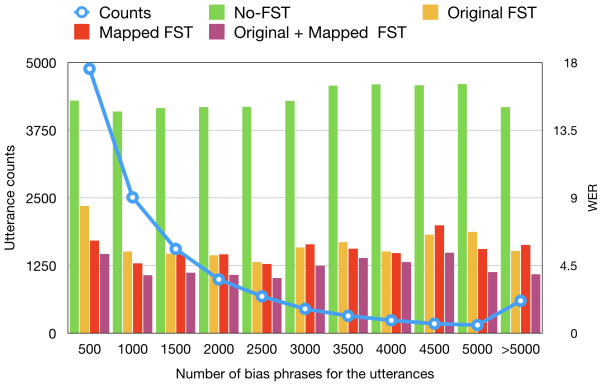

Figure 1: *Number of bias phrases in utterances and the WER breakdown by utterance buckets and configurations.*

Table 3: *Word mapping through pronunciation and its impact on the accuracy.*

Setup	Named entity set WER	Regular set WER
No context FST	15.3	5.1
FST from original words	6.5	5.2
FST from mapped words	5.5	5.2
Original+mapped FST	4.5	5.2

We use the configuration from setup 4 in Table 2 for the context FST. The WER reduction from context FST is still big (from 15.3 to 6.5, a 57.5% reduction). When using the subword sequence from the mapped words, the WER is reduced to 5.5. Interestingly, if using both the original sequence and mapped sequence in the FST, the WER is further reduced to 4.5. That's about 31% error reduction over original context FST. The overall WERR over the no-FST baseline is **70.6%**, with total inference time increased by 10%. Fig. 1 shows the WER breakdown in terms of number of bias phrases for the utterances. The WER reduction is consistent across utterances with different number of bias phrases.

We examine some mapped words in Table 4, the first column is the original word surface form, the second column is the mapped-to words through pronunciation, the third column is the recognized words when the original context FST is used. After using the subword sequence of the mapped word in the FST, these cases are fixed. From Table 4, we observe that rare words are mapped to more common words, either one-to-one map-

ping, or the original word is decomposed into multiple more common words, or less frequently, a phrase is reduced to a single word.

Table 4: *Sample words mapped through pronunciation. With the mapping, original words are correctly recognized.*

Original word	**Mapped-to word**	**Original recognition**
Yvanna	ivana	ivana
sista	sister	sister
Ellie Gershenwald	Elle Gershon walled	Ellie garcia walt
Vandendriessche	Vanden Drey Eske	vandendraci
La Juana	Lajuana	Louetta

6. Conclusions

User-specific content or location dependent content are challenging for E2E models since the model directly converts speech to text. In this study, we have empirically demonstrated that inserting class LM tags into the text for E2E model training is beneficial by itself. The contextual bias FST is a useful technique to inject external knowledge into the E2E model. To make this technique more accurate, we have proposed a simple score normalization algorithm. The accuracy improvement is 57%. We have also proposed to transform words through pronunciation. This algorithm converts the rare and unusual words into more common words so they are easier to recognize. This algorithm improves the accuracy by another 31%. The overall accuracy improvement over the no-FST baseline is 70.6%. Importantly, the proposed techniques cause little degradation on the non-named-entity type utterances. In the future, we plan to incorporate this word mapping process into training so the model is more accustomed to the sequences from both the original word and mapped-to word.

7. Acknowledgement

We would like to thank Matt Mirsamadi, Kyuyeon Hwang, Tim Ng, Roger Hsiao, Henry Mason, Leo Liu, Arnab Ghoshal, Yuchen Zhang, Man-Hung Siu, and John Bridle for helpful discussions.

8. References

[1] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural network," *Proc. ICML*, 2014.

[2] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *CoRR*, vol. abs/1508.01211, 2015. [Online]. Available: http://arxiv.org/abs/1508.01211

[3] A. Graves, "Sequence transduction with recurrent neural networks," *CoRR*, vol. abs/1211.3711, 2012. [Online]. Available: http://arxiv.org/abs/1211.3711

[4] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," *Proc. INTERSPEECH*, 2017.

[5] C.-C. Chiu, T. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. Weiss, K. Rao, K. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," *Proc. ICASSP*, vol. abs/1712.01769, 2018. [Online]. Available: http://arxiv.org/abs/1712.01769

[6] C. Raffel, T. Luong, P. J. Liu, R. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," *Proc. ICML*, vol. abs/1704.00784, 2017. [Online]. Available: http://arxiv.org/abs/1704.00784

[7] C. Chiu and C. Raffel, "Monotonic chunkwise attention," *Proc. ICLR*, vol. abs/1712.05382, 2018. [Online]. Available: http://arxiv.org/abs/1712.05382

[8] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Handbook of Speech Processing, J. Benesty, M. Sondhi, and Y. Huang, Eds.*, 2008, pp. 559–582.

[9] J. Novak, N. Minematsu, and K. Hirose, "Dynamic grammars with lookahead composition for wfst-based speech recognition," *Proc. INTERSPEECH*, 2012.

[10] P. Aleksic, M. Ghodsi, A. Michaely, C. Allauzen, K. Hall, B. Roark, D. Rybach, and P. Moreno, "Bringing contextual information to google speech recognition," *Proc. INTERSPEECH*, 2015.

[11] M. Paulik and R. Huang, "Method for supporting dynamic grammars in wfst-based asr," November 2016, uS Patent No. 9502031.

[12] G. Pundak, T. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep context: end-to-end contextual speech recognition," *Proc. SLT*, 2018.

[13] I. Williams, A. Kannan, P. Aleksic, D. Rybach, and T. Sainath, "Contextual speech recognition in end-to-end neural network systems using beam search," *Proc. INTERSPEECH*, 2018.

[14] D. Zhao, T. Sainath, D. Rybach, P.Rondon, D. Bhatia, B. Li, and R. Pang, "Shallow-fusion end-to-end contextual biasing," *Proc. INTERSPEECH*, 2019.

[15] Z. Chen, M. Jain, Y. Wang, M. Seltzer, and C. Fuegen, "End-to-end contextual speech recognition using class language models and a token passing decoder," *Proc. ICASSP*, 2019.

[16] A. Bruguier, R. Prabhavalkar, G. Pundak, and T. Sainath, "Phoebe: Pronunciation-aware contextualization for end-to-end speech recognition," in *Proc. ICASSP*, 2019.

[17] K. Hu, A. Bruguier, T. Sainath, R. Prabhavalkar, and G. Pundak, "Phoneme-based contextualization for cross-lingual speech recognition in end-to-end models," in *Proc. INTERSPEECH*, 2019.

[18] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1715–1725.

[19] K. Chen and Q. Huo, "Scalable training of deep learning machines by incremental block training with intra-block parallel optimization and blockwise model-update filtering," in *ICASSP*, March 2016.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. CVPR*, 2016.

[21] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. NIPS*, 2015, pp. 1171–1179.

[22] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, and et al., "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[23] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proc. INTERSPEECH*, 2019.

[24] R. Prabhavalkar, T. Sainath, Y. Wu, P. Nguyen, Z. Chen, C.-C. Chiu, and A. Kannan, "Minimum word error rate training for attention-based sequence-to-sequence models," in *Proc. ICASSP*, 2018, pp. 4839–4843.

[25] S. Shen, Y. Cheng, Z. He, W. He, H. Wu, M. Sun, and Y. Liu, "Minimum risk training for neural machine translation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1683–1692.