INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Approximate Nearest Neighbour Phrase Mining for Contextual Speech Recognition

Maurits Bleeker[1], Pawel Swietojanski[2], Stefan Braun[2], and Xiaodan Zhuang[2]*

[1]University of Amsterdam, The Netherlands
[2]Apple

m.j.r.bleeker@uva.nl, {pswietojanski, stefan_braun, xiaodan_zhuang}@apple.com

Abstract

This paper presents an extension to train end-to-end Context-Aware Transformer Transducer (CATT) models by using a simple, yet efficient method of mining hard negative phrases from the latent space of the context encoder. During training, given a reference query, we mine a number of similar phrases using approximate nearest neighbour search. These sampled phrases are then used as negative examples in the context list alongside random and ground truth contextual information. By including approximate nearest neighbour phrases in the context list during training, we encourage the learned representation to disambiguate between similar, but not identical, biasing phrases. This improves biasing accuracy when there are several similar phrases in the biasing inventory. We carry out experiments in a large-scale data regime obtaining up to 7% relative word error rate reductions for the contextual portion of test data. We also extend and evaluate CATT approach in streaming applications.
Index Terms: Neural Transducer, Contextual Speech Recognition, Approximate Nearest Neighbour sampling

1. Introduction

Recognizing words that are rare or unseen during training poses a challenge for end-to-end (E2E) automatic speech recognition (ASR) [1,2,3,4]. One way to address this problem is to allow the model to use user-specific information during inference, such as contact names, app names, media titles, and relevant geo-location names. To that end, several approaches have been proposed including shallow language model (LM) fusion [5,6], on-the-fly rescoring [5,7,8], or deep fusion approaches [3,9,10]. As E2E models tend to learn a strong internal LM [11,12], shallow LM fusion and rescoring approaches are not always effective out of the box.

Alternative methods rely on deep neural contextual fusion (DCF) [3,9,10,13,14,15,16]. In DCF, the biasing machinery is part of the ASR model and is jointly learned with the main ASR objective. Different DCF techniques share much of the same modeling back-end and thus can be implemented for arbitrary E2E network architectures such as the attention encoder-decoder (AED) [17] or the neural transducer [18] (RNN-T) ASR systems. Deep contextual biasing has been proposed for the Contextual Listen, Attend, and Spell (CLAS) [9,17] AED model, and similar solutions were extended to the contextual neural transducers [10, 13]. The major difference between contextual models and their non-contextual counterparts is the biasing machinery, usually implemented as an additional context encoder followed by a fusion mechanism. The context encoder is typically implemented as an LSTM [19] or more recently a transformer [20] model, and its role is to project a set of tokenized biasing phrases into a set of fixed-sized continuous embeddings. Next, a fusion mechanism integrates these embeddings with the acoustic (AED, RNN-T) and/or label (RNN-T) encoder when making ASR predictions. Fusion can be implemented in a latent space with cross-attention between audio and context encoders [9,10] or by interpolating generic and contextual model's distributions, as done in tree-constrained pointer generation networks (TCPGN) [15].

A major challenge in contextual biasing is that some words, including the biasing phrases fed into the context encoder, may exhibit phonetic similarities with one another or may be characterized by complex and non-standard pronunciation patterns. For example, names in a contact list that sound similar to each other, or geo-location names that have similar (but not identical) pronunciations. To make deep contextual biasing more robust to settings where context information is (phonetically) similar to each other, one could explicitly embed additional phoneme-level information in the contextual ASR as explored in [21], or train the ASR system such that it can learn to better disambiguate between challenging queries.

In this work, we are interested in the latter approach by exposing the ASR to hard negative examples during training. Alon et al. [4] proposed a method to generate phonetically similar phrases given a reference phrase. Phonetically similar phrases might have similar (i.e., confusing) acoustic representations and, therefore, are hard to distinguish from the desired phrase during inference [4]. By appending phonetically similar phrases as *hard negatives* to the context encoder's inputs during training, the model is explicitly tasked to disambiguate between them. There are several possible ways to insert hard negatives into the training pipeline. In [4], an external ASR model [22] is used to decode and generate a set of hypotheses for each query. These hypotheses are then ranked based on the word co-occurrence and the phonetic similarity with the reference phrase.

While the method by Alon et al. [4] has shown promising results, it is worth noting that their approach may be viewed as a form of data augmentation implemented prior to training of a deep contextual ASR model, rather than a technique that can be directly integrated into the training process. This may not be optimal, as exact hard negative phrases (HNP) are likely to depend on the mistakes a specific ASR is prone to make, rather than mis-recognitions of some independent ASR system.

In this paper, we present an alternative, computationally efficient extension of mining HNP: approximate nearest neighbour phrases (ANN-P) mining. ANN-P mining allows to efficiently select HNP during the training of a deep contextual model in an online manner, using the latent space of the context encoder. ANN-P mining unifies two important aspects of HNP for contextual ASR in a single method, by including phrases in the context list that are (i) phonetically similar to the reference phrase (i.e.,

*Work done while interning at Apple.

 10.21437/Interspeech.2023-2136

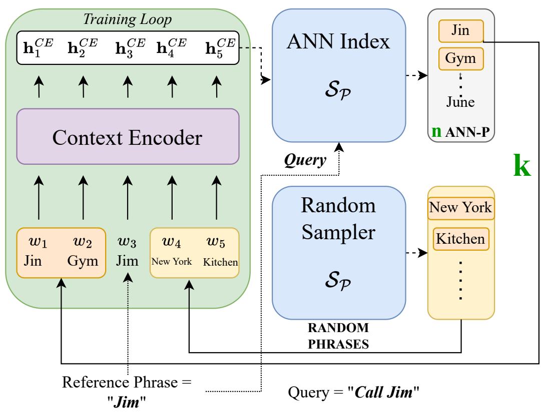

Figure 1: *Overview of the ANN-P mining method. Given a query* ***"Call Jim"****, we append k ANN phrases to the context list. Given a query phrase, we first sample* n *phrases from the ANN index. Next, we sample the top* k*. The remaining phrases in the context list are randomly sampled.*

distracting phrases), as in [4], and (ii) close to the reference phrase in the latent space of the context encoder. The latter property is hypothesized to make it harder for the model to discriminate different phrases [4]. Different from Alon et al. [4], our approach does not require a full decode of training data, nor an existing pre-trained ASR model to obtain hard negatives.

We implement the proposed ANN-P mining using the Context-Aware Transformer Transducer (CATT) model [13]. CATT proposed an efficient biasing approach that makes full use of the Transformer Transducer (TT) [23] architecture. The biasing phrases in CATT are mapped into single embeddings (one per biasing phrase) that is then used with cross-attention to bias both audio and label encoders. While the original CATT formulation only considered non-streaming scenarios, we extend CATT to both streaming and non-streaming applications at the same time by training it in a variable attention masking manner [24, 25]. Given the limited (audio) context window in streaming scenarios, it is difficult to accurately distinguish the correct biasing phrase while transcribing. This especially applies for CATT-like approaches where each biasing phrase is compressed into a single embedding, rather than an embedding per sub-word as is the case with TCPGN or neural associative networks (NAM) [16] [1]. As such, we anticipate that for the CATT model, ANN-P mining is likely to provide greater value in streaming than in non-streaming scenarios.

The contributions of this work thus include:

- An extension of the training mechanism for CATT that utilizes mining hard negative phrases directly from the latent space of the ASR model. This approach results in up to 7% relative word error rate reductions for the personalized portion of test data, with only minor regressions on generic queries.
- An extension of CATT to the streaming scenario.
- Evaluation of the proposed approach in a large-scale data regime consisting of 650,000 hours of acoustic training data. This is to make sure the models are well-trained, and not handicapped by limited training data diversity, which in turn could artificially inflate biasing performance.

[1]Although not investigated in this work, CATT is likely to work well for neural biasing of non-auto-regressive models like connectionist temporal classification (CTC) [26], whereas NAM or TCPGN relies on decoded prefixes for biasing. See [27] for a recent CTC study.

2. Method

2.1. Context-Aware Transformer Transducer

CATT [13] extends an RNN-T consisting of a label and audio encoder, with an additional contextual encoder, followed by a biasing cross-attention layer that measures the relevance of the context phrase to the query from the perspective of information found in audio and label encoders. The context encoder $f^{context}(\cdot)$ takes as input a set of context phrases $\mathcal{S}_{\mathcal{C}} = \{w_1, \dots\}$ and maps each context phrase $w_i \in \mathcal{S}_{\mathcal{C}}$ into a fixed vector representation $f^{context}(w_i) := \mathbf{h}_i^{CE}$. CATT uses a transformer [20] for the label, audio, and context encoder.

The biasing layer in the CATT model consists of a multi-head attention (MHA) layer [20]. The goal of the MHA is to measure the similarity between each phrase in the context list and the audio signal using scaled dot-product cross-attention, which weights each input phrase according to this similarity score with the audio.

Consider a pair of contextual phrases $w_i, w_j \in \mathcal{S}_{\mathcal{C}}$, where $i \neq j$. Phrase w_i is the reference phrase in the audio signal and w_j is considered as a negative phrase w.r.t. the audio transcript. Phrases i and j have close to each other (key) embeddings ($\mathbf{k}_i$ and $\mathbf{k}_j$ respectively), in terms of a similarity metric. When a query-key pair has a low attention score, the matching value embeddings get close to zero weight score. Since $\mathbf{k}_i$ and $\mathbf{k}_j$ are close to each other, $\mathbf{k}_j$ may be distracting for the attention head(s) yielding too high attention score, hence the distracting phrase may be taken into account when transcribing the audio.

2.2. Proposed method: ANN-P mining

In this section, we present an approach to extend the training of the CATT by introducing ANN-P mining. The goal of ANN-P mining is to select phrases similar to the *reference phrase* in terms of their similarity in the latent space of the context encoder. When mining the phrases randomly, the probability of having (phonetically) similar phrases in the context list is negligible. Therefore, the label, audio, and context encoder may not learn to disambiguate between similar sounding (i.e., difficult to discriminate) phrases. In Fig. 1 we provide a high-level overview of our ANN-P mining method.

Prior to training, we extract all the biasing phrases from each audio transcription using the existing automatically generated meta-information on entity spans. This results in a set of phrases for the entire training data $\mathcal{S}_{\mathcal{P}}$, referred to as the biasing phrases inventory. $\mathcal{S}_{\mathcal{P}}$ can be extended with additional entries that are not present in the training data to provide additional context, as needed.

The goal of ANN-P mining is to select hard negative samples according to the current state of the trained ASR model. We use the context encoder of the CATT to encode each phrase $w_i \in \mathcal{S}_{\mathcal{P}}$ into its latent representation $\mathbf{h}_i^{CE}$ given a checkpoint of the CATT during training (left box in Fig. 1). We cache the latent representation $\mathbf{h}_i^{CE}$ of each phrase into an online approximate nearest neighbour (ANN) index (i.e., an index that can be efficiently queried during training).

Given a *query phrase* w_i, the ANN index maps the phrase w_i to its cached latent representation $w_i \rightarrow \mathbf{h}_i^{CE}$ and, using ANN search over all cached phrase embeddings, returns n ANN-P from the index based on the dot product score w.r.t. the query phrase. To prevent sampling the same phrases at every epoch for the same query, we randomly sample k (where $k < n$) phrases from the n retrieved phrases. The remaining phrases (if any) that are needed for the given query are added randomly by sampling from $\mathcal{S}_\mathcal{P}$. Together the ANN-P and the randomly sampled phrases are included in the context list. ANN-P mining can only be used if the query phrase is present in the ANN index since we need its neural representation to apply similarity search. Hence, we need to cache each phrase first before we can apply ANN-P mining. However, an indexing step is an inexpensive process, taking a small percentage of the total training time.

For ANN-P mining, we need one representation per biasing phrase (all from the same latent space) to store in the ANN index. In this work, we use the output representation of the context encoder $\mathbf{h}_i^{CE}$ for ANN-P mining. However, CATT measures the similarity between the audio and the phrases by using a MHA biasing layer, which uses 8 attention heads. Hence, there are 8 different phrase representations (living in different latent spaces) that are used to compute an attention score (i,e, similarity) between a phrase and the audio signal. There are several ways to aggregate the 8 key representations into a single representation that could be used for approximate nearest neighbour search. As a straightforward approach that applies as an approximation for all the 8 key representations, we store the output representation of the context encoder $\mathbf{h}_i^{CE}$ in the index instead (this is because the key projection is only a linear transformation).

3. Experimental Setup

We carry out the experiments on a large-scale dataset consisting of queries from two tasks: dictation and assistant. The semi-supervised portion of the data consists of around 600,000 hours of randomized and anonymized automatically transcribed acoustic data, while the supervised part contains around 50,000 hours of randomized and anonymized English queries.

Following [25, 28], our systems are trained in a two-stage manner - the first stage pre-trains the models on semi-supervised data for a total of 5.6M updates, the second fine-tunes the model for another 280k updates on supervised data. In both stages, gradients are accumulated over 9216 queries. We use SyncSGD + Adam [29] for distributed optimization, with exponentially decaying learning rates. ANN-P sampling is applied only in the fine-tuning stage and not during evaluation. All models are evaluated using a test set containing 60 hours of assistant data. Around 40% of the test set consists of contextual queries spanning domains such as contact, app, and geo-location names. During inference we include real user profiles in the context list. The remainder consists of queries that are generic in nature and are unlikely to benefit from personalized priors.

3.1. Contextual Transformer Transducer Model

In this work, our base contextual E2E ASR model is the Context-Aware Transformer Transducer (CATT) [13], configured to have around 120M parameters. The audio encoder is a 12-layer Conformer [30] while the label and context encoders are implemented as a 6-layer transformer model. Each encoder has an embedding size of 512 and the MHA is configured to 8 attention heads. All examples in the training batch share the same context list, which allows exposing each query to a larger number of context phrases, at the same time keeping the memory usage low. To do so, we only sample a few random + ANN-P per query and combine them into a single context list for all queries in a mini-batch. This was configured such that each query has access to around 96 - 128 biasing phrases.

Different from the original CATT, we append a back-off phrase to the context list where the model can attend to in case there are no relevant-biasing phrases. Adding a back-off token has also been demonstrated to be effective with CLAS [9]. Originally, CATT was only trained and evaluated with global context models. In this work, we also investigate its suitability to both non-streaming and streaming applications. We do so by training both CATT and baseline models in variable masking manner [24, 25], and then configuring the models to either streaming or non-streaming settings during decoding. Streaming models operate on 240ms long causal audio chunks [31, 32] and thus have limited access to the future audio signal which may pose a challenge for CATT approach.

To show performances without biasing machinery, we compare with a Transformer Transducer (TT) [23] that has the same audio and label encoder architecture as CATT and has been trained with multiple attention masks to allow for streaming and non-streaming decoding. The exact architecture details of the TT can be found in [25]. Since streaming models are expected to emit tokens with low partial latency, we train both TT and CATT models with the latency-penalizing FastEmit loss [33].

3.2. ANN index and negative phrase mining

The ANN-P index is built using the Annoy[2] library. For the ANN-P mining, we experiment with various ways of mixing negative examples into context lists. In general, we append 8 biasing phrases for each query[3] , where some proportion is expected to be made of ANN-P (see section 4.2 for details), while the remainder is randomly selected out of the biasing phrases inventory. We use the dot product between phrase embeddings and a query as similarity metric to mine ANN-P. Given a training query, and its corresponding reference biasing phrase(s) (if any), we retrieve *n* approximate nearest neighbours form the index. Important to notice, we sample each negative phrase at the word level (i.e., if a query phrase consists of multiple words, we sample n HNPs per word). Next, we sample *k* phrases at random from the retrieved *n* ANNs. After completing two fine-tuning epochs, the ANN index is rebuilt by re-indexing the phrase representations using the latest state of context encoder parameters. To prevent over-fitting on the same ANN-P, we do not sample ANN-P for every query in the training batch but use them proportionally to the *append ratio*. The frequency of adding ANN-P to the context list during training increases as the append ratio increases. In cases where we are unable to sample ANN-Ps for a query, we use random phrases instead.

4. Results

4.1. Random vs ANN-sampled phrases

Table 1 reports the results for models operating in global (i.e., non-streaming) (upper block) and streaming (lower block) modes, with and without access to contextual information. To demonstrate the effect on Word Error Rate (WER) in situations where contextual information is absent, we also evaluate CATT without access to relevant contextual information. In this work, global and streaming models are the same models, configured to different operating regimes via different settings of attention masking [25]. Note that [13] only investigated the biasing performance of CATT in a global setting, and thus it is unclear if and to what extent the CATT approach can be used in streaming applications.

[2] `https://github.com/spotify/annoy`

[3] Note, we eventually share these across all examples in the batch, so for a batch of 16 queries and 8 contextual phrases per query, each query would make use of 128 biasing phrases to pick from.

Model	Ctx. Info	WER [%] Generic	Personal	Avg.
		Global decoding		
TT	-	3.9	11.9	6.3
CATT	✗	4.3	11.5	6.5
CATT w/ ANN-P	✗	4.4	11.5	6.5
CATT	✓	4.5	2.6	3.9
CATT w/ ANN-P	✓	4.5	2.6	3.9
		Streaming decoding		
TT	-	4.5	12.3	6.8
CATT	✗	4.6	11.4	6.7
CATT w/ ANN-P	✗	4.5	11.6	6.7
CATT	✓	4.9	2.8	4.3
CATT w/ ANN-P	✓	4.9	**2.6**	**4.2**

Table 1: *WERs for models configured to global (upper block) and streaming (lower block) decodings, with and without access to contextual information (Ctx. Info). CATT models are trained with random, or ANN-P sampling. The results on test data are additionally aggregated on generic and personalized subsets. The Generic portion consists of queries that are not expected to benefit from contextual information. Results obtained for n=20, k=2, and append ratio=0.25.*

When decoding CATT models in global mode, allowing the model to access contextual information during inference improves accuracy by 38% relative WER (WERR) on average (i.e., 6.5% vs. 3.9% for non-biased and biased CATT systems, respectively). This is accompanied by a 3% WERR degradation on a non-personal (generic) portion of the test set (i.e., 6.3% vs. 6.5% on average for TT and CATT, respectively). These results are in line with findings on CATT and global decodes reported in [13]. Another observation is that training with ANN-P mining does not seem to affect the non-streaming results in a significant way. This can be most likely explained by the fact that having access to the entire audio sequence allows the model to better contextualize the information, thus it is easier to match the complete audio evidence to the correct biasing phrase.

When decoding CATT models configured to streaming mode, we observe similar overall trends as with the global models. Interestingly, for the streaming scenario, the regression on WER for the non-contextual portion of data no longer exists when compared to a baseline TT model (i.e., 6.8% vs 6.7% for the baseline and CATT, respectively). For the streaming scenario, we also obtain up to 7% WERR reductions on the contextual portion of the test set (this is where ANN-P is expected to help).

We can conclude that for the limited audio look-ahead, training CATT with ANN-P mining helps to improve accuracy by allowing the model to better disambiguate between contextual phrases. Since in CATT the biasing information is compressed into a single embedding, the use of HNP helps to regularize embeddings such they are more robust to small phonetic variations.

4.2. Further analyses

ANN-P mining has three main hyper-parameters: n (i.e, the number of ANN-P we take from the index), k (i.e., the k phrases we sample from the n ANNs), and the *append ratio* (how frequent we add the ANN-P to the context list). We depict the effect of each parameter in Fig. 2. We can conclude that, in general, ANN-P mining is robust to the choice of the considered hyper-parameters. The lowest scores are obtained for an append-ratio of 1 and using $n = 20$ phrases from the index. The number of phrases (k) appended to the context list does not seem to have a

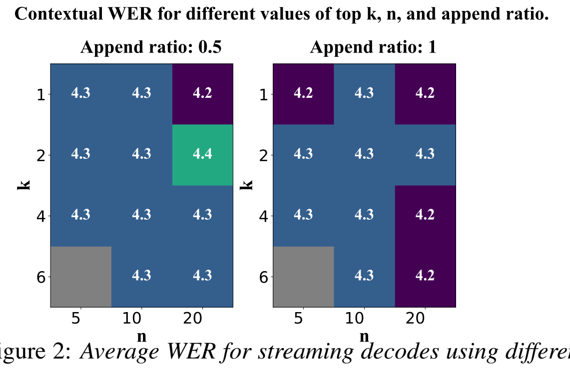

Figure 2: *Average WER for streaming decodes using different values of top k, n, and append ratio.*

Query Phrase	$n = 4$ ANN Phrases
john	'joan', 'johnson', 'johann', 'from john'
building	'buildings', 'builder', 'the building', 'builds'
jean	'jeanne', 'jeannie', 'jeana', 'jeanine'
eva	'evie', 'ava', 'evin', 'evy'
play	'playa', 'place', 'flay', 'platte'

Table 2: *Query Phrases and their ANNs retrieved from the index.*

strong effect on the WER.

In this work, we mine hard negative phrases (at the word level) from the latent space of the context encoder, based on the neural similarity with the reference (i.e., query) phrase. In Table 2, we provide several examples of query phrases and their top four ANN-P as retrieved from the index. We can observe that top ANN-P mainly results in phrases that are phonetically similar to the query phrase. For queries consisting of names, we mainly retrieve other similar-sounding (but different) names. For a query such as *building*, we mainly retrieve ANN-P that are related to the same concept and contain the sub-word *build*.

Finally, we also investigated the following aspects and report them here for completeness. These experiments either did not significantly impact accuracy or led to deterioration:

- Rebuilding the ANN-P index at different epochs did not have a significant impact on WER.
- Enabling ANN-P mining at different stages of training, including the pre-training, or fine-tuning the last 2-3 epochs did not improve over using it during the entire fine-tuning stage.
- Sampling ANN-P using multi-word phrases, instead of single-word phrases, resulted in 5% WERR degradation.

5. Discussion & Conclusion

We proposed and evaluated an efficient method for mining approximate nearest neighbour phrases (i.e., hard negatives) for transformer-transducer contextual speech recognition based on CATT model. In order to mine hard negatives, our method does not require an external ASR model, nor additional decodings of training data. We also extended CATT modelling to streaming applications by training it with multiple attention mask configurations. We evaluated the proposed ideas in large-scale data experiments, finding that the CATT using the ANN-P mining approach offers up to 7% relative WER reductions for streaming models on the personalized portion of the test data.

6. Acknowledgements

We thank Dogan Can, Nikos Flemotomos, Roger Hsiao, Vijay Peddinti, Honza Silovsky, and Russ Webb for their valuable suggestions on this work.

7. References

[1] J. Guo, T. N. Sainath, and R. J. Weiss, "A spelling correction model for end-to-end speech recognition," in *Proc. ICASSP*. IEEE, 2019, pp. 5651–5655.

[2] T. N. Sainath, R. Prabhavalkar, S. Kumar, S. Lee, A. Kannan, D. Rybach, V. Schogol, P. Nguyen, B. Li, Y. Wu, Z. Chen, and C. Chiu, "No need for a lexicon? evaluating the value of the pronunciation lexica in end-to-end models," in *Proc. ICASSP*. IEEE, 2018, pp. 5859–5863.

[3] A. Bruguier, R. Prabhavalkar, G. Pundak, and T. N. Sainath, "Phoebe: Pronunciation-aware contextualization for end-to-end speech recognition," in *Proc. ICASSP*. IEEE, 2019, pp. 6171–6175.

[4] U. Alon, G. Pundak, and T. N. Sainath, "Contextual speech recognition with difficult negative training examples," in *Proc. ICASSP*. IEEE, 2019, pp. 6440–6444.

[5] D. Zhao, T. N. Sainath, D. Rybach, P. Rondon, D. Bhatia, B. Li, and R. Pang, "Shallow-fusion end-to-end contextual biasing-williams2018contextual," in *Proc. Interspeech*. ISCA, 2019, pp. 1418–1422.

[6] D. Le, G. Keren, J. Chan, J. Mahadeokar, C. Fuegen, and M. L. Seltzer, "Deep shallow fusion for rnn-t personalization," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 251–257.

[7] K. B. Hall, E. Cho, C. Allauzen, F. Beaufays, N. Coccaro, K. Nakajima, M. Riley, B. Roark, D. Rybach, and L. Zhang, "Composition-based on-the-fly rescoring for salient n-gram biasing," in *Proc. Interspeech*. ISCA, 2015, pp. 1418–1422.

[8] I. Williams, A. Kannan, P. S. Aleksic, D. Rybach, and T. N. Sainath, "Contextual speech recognition in end-to-end neural network systems using beam search," in *Proc. Interspeech*. ISCA, 2018, pp. 2227–2231.

[9] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep context: End-to-end contextual speech recognition," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 418–425.

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

[11] E. McDermott, H. Sak, and E. Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," in *Automatic Speech Recognition and Understanding Workshop, ASRU*. IEEE, 2021, pp. 434–441.

[12] Z. Meng, N. Kanda, Y. Gaur, S. Parthasarathy, E. Sun, L. Lu, X. Chen, J. Li, and Y. Gong, "Internal language model estimation for domain-adaptive end-to-end speech recognition," in *Proc. ICASSP*. IEEE, 2021, pp. 7338–7342.

[13] F. Chang, J. Liu, M. Radfar, A. Mouchtaris, M. Omologo, A. Rastrow, and S. Kunzmann, "Context-aware transformer transducer for speech recognition," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 503–510.

[14] K. M. Sathyendra, T. Muniyappa, F. Chang, J. Liu, J. Su, G. P. Strimel, A. Mouchtaris, and S. Kunzmann, "Contextual adapters for personalized speech recognition in neural transducers," in *Proc. ICASSP*. IEEE, 2022, pp. 8537–8541.

[15] G. Sun, C. Zhang, and P. C. Woodland, "Tree-constrained pointer generator for end-to-end contextual speech recognition," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 780–787.

[16] T. Munkhdalai, K. C. Sim, A. Chandorkar, F. Gao, M. Chua, T. Strohman, and F. Beaufays, "Fast contextual adaptation with neural associative memory for on-device personalized speech recognition," in *Proc. ICASSP*. IEEE, 2022, pp. 6632–6636.

[17] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[18] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, pp. 1735–1780, 1997.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeuRIPS*, 2017, pp. 5998–6008.

[21] Z. Chen, M. Jain, Y. Wang, M. L. Seltzer, and C. Fuegen, "Joint grapheme and phoneme embeddings for contextual end-to-end asr." in *Proc. Interspeech*. ISCA, 2019, pp. 3490–3494.

[22] E. Variani, T. Bagby, E. McDermott, and M. Bacchiani, "End-to-end training of acoustic models for large vocabulary continuous speech recognition with tensorflow," in *Proc. Interspeech*. ISCA, 2017, pp. 1641–1645.

[23] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer Transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *Proc. ICASSP*. IEEE, 2020, pp. 7829–7833.

[24] A. Tripathi, J. Kim, Q. Zhang, H. Lu, and H. Sak, "Transformer transducer: One model unifying streaming and non-streaming speech recognition," *arXiv preprint arXiv:2010.03192*, 2020.

[25] P. Swietojanski, S. Braun, D. Can, T. F. da Silva, A. Ghoshal, T. Hori, R. Hsiao, H. Mason, E. McDermott, H. Silovsky *et al.*, "Variable attention masking for configurable transformer transducer speech recognition," in *Proc. ICASSP*. IEEE, 2023.

[26] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural nets," in *Proc. ICML*, 2006.

[27] S. Dingliwal, M. Sunkara, S. Ronanki, J. Farris, K. Kirchhoff, and S. Bodapati, "Personalization of ctc speech recognition models," in *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 302–309.

[28] T. Nguyen, N. Tran, L. Deng, T. F. da Silva, M. Radzihovsky, R. Hsiao, H. Mason, S. Braun, E. McDermott, D. Can *et al.*, "Optimizing bilingual neural transducer with synthetic code-switching text generation," *arXiv preprint arXiv:2210.12214*, 2022.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2014.

[30] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*. ISCA, 2020, pp. 5036–5040.

[31] Y. Shi, Y. Wang, C. Wu, C. Yeh, J. Chan, F. Zhang, D. Le, and M. Seltzer, "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition," in *Proc. ICASSP*. IEEE, 2021, pp. 6783–6787.

[32] X. Chen, Y. Wu, Z. Wang, S. Liu, and J. Li, "Developing real-time streaming transformer transducer for speech recognition on large-scale dataset," in *Proc. ICASSP*. IEEE, 2021, pp. 5904–5908.

[33] J. Yu, C. Chiu, B. Li, S. Chang, T. N. Sainath, Y. He, A. Narayanan, W. Han, A. Gulati, Y. Wu, and R. Pang, "Fastemit: Low-latency streaming ASR with sequence-level emission regularization," in *Proc. ICASSP*. IEEE, 2021, pp. 6004–6008.