INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Dual-Mode NAM: Effective Top-K Context Injection for End-to-End ASR

Zelin Wu, Tsendsuren Munkhdalai, Pat Rondon, Golan Pundak, Khe Chai Sim, Christopher Li

Google LLC, USA

{zelinwu, tsendsuren, rondon, golan, khechai, chriswli}@google.com

Abstract

ASR systems in real applications must be adapted on the fly to correctly recognize task-specific contextual terms, such as contacts, application names and media entities. However, it is challenging to achieve scalability, large in-domain quality gains, and minimal out-of-domain quality regressions simultaneously. In this work, we introduce an effective neural biasing architecture called Dual-Mode NAM. Dual-Mode NAM embeds a top-k search process in its attention mechanism in a trainable fashion to perform an accurate top-k phrase selection before injecting the corresponding wordpiece context into the acoustic encoder. We further propose a controllable mechanism to enable the ASR system to be able to trade off its in-domain and out-of-domain quality at inference time. When evaluated on a large-scale biasing benchmark, the combined techniques improve a previously proposed method with an average in-domain and out-of-domain WER reduction by up to 53.3% and 12.0% relative respectively.

Index Terms: speech recognition, contextual adaptation

1. Introduction

Rare words such as person and application names and titles from a media library can be challenging for Automatic Speech Recognition (ASR) systems to transcribe, as they are often specific to an individual user and may only be available at inference time. On-device personalization of ASR models can help acquiring new phrases [1, 2, 3], but personalization requires training data in the form of audio samples paired with their transcriptions; the need to train on paired data limits applicability and introduces a delay before new words can be recognized. On the other hand, contextual biasing techniques can learn from text alone and support immediate vocabulary acquisition.

There is growing interest in end-to-end neural biasing to improve the performance of ASR on contextual long-tail words [4, 5, 6, 7]. Unlike previous methods that combine an external language model (LM) with ASR system in a heuristic way to rescore hypotheses as they are added to the lattice [8, 9], end-to-end neural biasing incorporates a biasing module into the ASR model itself and performs edits in the latent feature space to adapt to a new unseen vocabulary. This close integration enables joint optimization of the neural biasing module and the ASR system and allows for building a biasing component in a data-driven fashion; the resulting models show competitive, or even improved, recognition performance compared to Finite State Transducer (FST) LM methods when biasing contextual phrases of varying types. End-to-end neural biasing approaches can also leverage accelerator devices (e.g. GPU and TPU) and support batch inference for reduced latency.

This work builds on the Neural Associative Memory (NAM) [10] framework. Unlike prior approaches, where the biasing attention is computed over phrase-level embeddings, NAM directly attends to wordpiece (WP) embeddings for fine-grained contextual biasing. NAM also builds a structured memory that stores bindings between the current and next WPs as a key-value pair. The use of an attention layer allows NAM to consume a variable-sized set of contextual phrases; the run time of this attention layer, and the need for ASR to operate in real time, then becomes a limit on how large the inference-time context can be. There is a long line of work on improving the efficiency of attention in the context of sequence-to-sequence modeling [11, 12, 13]; FineCoS [14] and NAM+ [15] improve the efficiency of attention in neural biasing by using of phrase-level embeddings to narrow the set of token-level embeddings used when computing the final context embedding. NAM+ [15] in particular scales to large numbers of contextual phrases using Two-Pass Hierarchical Attention (THA), in which a summary representation of each phrase is first built from the embeddings of its constituent WPs, and then a top-k search selects the K most-similar phrases before applying WP-level NAM attention. While THA improves inference speed nearly 16-fold, training only WP attention while reusing (a summary of) the learned WP embeddings for phrase-level attention at inference creates a train-test mismatch, degrading WER. Further, the NAM and NAM+ training process ties the size of the training-time synthetic context to the training batch size, making it impossible to make the synthetic context smaller, to reduce training time and difficulty, or larger, to improve out-of-domain accuracy.

In this work, we introduce Dual-Mode NAM, which improves on NAM+ in three ways. First, Dual-Mode NAM learns discrete phrase- and WP-level embeddings and attention, where the phrase and WP attention networks are trained in parallel, eliminating NAM+'s train-test mismatch in how phrase-level attention and embeddings are learned and applied. Second, Dual-Mode NAM introduces a parameter that allows for dynamic, inference-time-control of biasing strength to trade off in-domain and out-of-domain accuracy. Finally, Dual-Mode NAM introduces a training-time synthetic context construction strategy that exposes the model to more-diverse data and allows the size of the per-example synthetic context to vary independently of batch size. Together, these changes to Dual-Mode NAM improve the previously proposed NAM+ with an average in-domain and out-of-domain Word Error Rate (WER) reduction by up to 53.3% and 12.0% relative, respectively.

2. Methods

In this section, we show how Dual-Mode NAM extends NAM+ with discrete, learned phrase- and WP-level embeddings and attention; improved training-time synthetic context generation; and inference-time-controllable biasing strength.

10.21437/Interspeech.2023-1124

2.1. Dual-Mode NAM training

2.1.1. Context encoder

We denote the set of input WP id sequences to the context encoder as $Z = \{cls; w_{n,1}, ..., w_{n,L}\}_{n=1}^{N}$, where $w_{n,1}$ denotes the start of sequence symbol <S>, L denotes the number of WPs for each bias phrase (padded with the end of sequence symbol </S> up to L); N denotes the number of bias phrases assigned to a given speech utterance; and, similarly to [16, 17], the cls id is prepended to help the encoder summarize phrase-level representations. The context encoder encodes Z and splits the output into phrase embeddings $E^p \in \mathbb{R}^{d \times N}$ and WP embeddings $E^w \in \mathbb{R}^{d \times N \times L}$, where d denotes the embedding size:

$$E^p, E^w = ContextEncoder(Z) \tag{1}$$

2.1.2. Wordpiece-mode attention

The WP attention adapts the NAM multi-headed attention [10], which models the transition from the current word-pieces E^w (flattened as $\mathbb{R}^{dNL}$) to the next word-pieces E^{w_s} (equivalent to E^w shifted to the left by one WP and padded with a zero embedding e^{zero} at the end). For example, if the bias phrases are [Lego House, photograph] and $L = 4$, E^w would correspond to [<S>, _Lego, _House, </S>, <S>, _photo, g, raph], and E^{w_s} would correspond to [_Lego, _House, </S>, e^{zero}, _photo, g, raph, e^{zero}]. We modified it with two optimizations:

1. W^{O_w} projects the intermediate context to the same dimension of the acoustic feature x_t without additional projection.
2. Learnable no-bias token $(e_h^{w\_nb\_k}, e_h^{w\_nb\_v})$ is concatenated to (K_h^w, V_h^w), not (E^w, E^{w_s}), skipping input projections. According to [4, 18], the no-bias token can improve the overall quality, by enabling the model to learn not to bias when there's a mismatch between the audio and biasing context.

$$x_t^w = FeedForward^w(x_t) \tag{2}$$

$$E^{w\_s} = \{e_{n,2}^w, ..., e_{n,L}^w, e^{zero}\}_{n=1}^{N} \tag{3}$$

$$q_{t,h}^w = x_t^w W_h^{Q_w}, K_h^w = E^w W_h^{K_w}, V_h^w = E^{w\_s} W_h^{V_w} \tag{4}$$

$$K_h^{w\_e} = [e_h^{w\_nb\_k}; K_h^w], V_h^{w\_e} = [e_h^{w\_nb\_v}; V_h^w] \tag{5}$$

$$\sigma_{t,h}^w = softmax(\frac{q_{t,h}^w K_h^{w\_e}}{\sqrt{d_k^w}}), c_{t,h}^w = \sigma_{t,h}^w V_h^{w\_e} \tag{6}$$

$$c_t^w = Concat(c_{t,1}^w, ..., c_{t,H}^w) W^{O_w} \tag{7}$$

Where t denotes the time index, $h \in [1..H]$ denotes the attention head index; in addition to $Feedforward^w$, the trainable parameters include $W_h^{Q_w} \in \mathbb{R}^{d \times d_k^w}$, $W_h^{K_w} \in \mathbb{R}^{d \times d_k^w}$, $W_h^{V_w} \in \mathbb{R}^{d \times d_v^w}$, $e_h^{w\_nb\_k} \in \mathbb{R}^{d_k^w}$, $e_h^{w\_nb\_v} \in \mathbb{R}^{d_v^w}$, $W^{O_w} \in \mathbb{R}^{Hd_v^w \times d_o}$.

2.1.3. Phrase-mode attention

To enable the ASR model to effectively narrow down the WP embeddings for WP attention during inference, a separate single-headed attention network is trained to attend to the bias phrases E^p (flattened as $\mathbb{R}^{dN}$), where the attention context c_t^p is used during training and can be discarded during inference.

$$x_t^p = FeedForward^p(x_t) \tag{8}$$

$$q_t^p = x_t^p W^{Q_p}, K^p = E^p W^{K_p}, V^p = E^p W^{V_p} \tag{9}$$

$$K^{p\_e} = [e^{p\_nb\_k}; K^p], V^{p\_e} = [e^{p\_nb\_v}; V^p] \tag{10}$$

$$c_t^p = softmax(\frac{q_t^p K^{p\text{-}e}}{\sqrt{d_k^p}}) V^{p\text{-}e} W^{O_p} \tag{11}$$

In addition to $Feedforward^p$, the trainable parameters include $W^{Q_p} \in \mathbb{R}^{d \times d_k^p}$, $W^{K_p} \in \mathbb{R}^{d \times d_k^p}$, $W^{V_p} \in \mathbb{R}^{d \times d_v^p}$, $e^{p\_nb\_k} \in \mathbb{R}^{d_k^p}$, $e^{p\_nb\_v} \in \mathbb{R}^{d_v^p}$ and $W^{O_p} \in \mathbb{R}^{d_v^p \times d_o}$.

2.1.4. Sampling-based dual-mode training

We applied a within-batch sampling strategy for dual mode training, i.e., for every mini-batch, a fraction of utterances (p) are trained with the phrase context c_t^p while the rest of the utterances ($1-p$) are trained with the WP context c_t^w. This concept is similar to sampling streaming / non-streaming encoders in [19].

$$c_t = \begin{cases} c_t^p & \text{with probability } p \\ c_t^w & \text{with probability } 1 - p \end{cases} \tag{12}$$

$$x_t^{biased} = x_t + c_t \tag{13}$$

Following NAM [10], the attention context c_t is added to the acoustic features x_t to produce the biased features x_t^{biased}.

2.2. Dual-mode NAM inference

2.2.1. Improved two-pass hierarchical attention (THA)

The prior THA [15] proposes filtering top-k (k^p) bias phrases before applying the WP attention, and it approximates the phrase search using only the WP attention during inference:

1. The phrase embeddings E^p are approximated by summing the WP embeddings for each phrase during inference.
2. The WP attention is reused for top-k phrase search, where $K_h^w = E^p W_h^{K_w}$ and H is the number of attention heads.

$$indices_t^{p\_topk} = TopK(\frac{1}{H} \sum_{h=1}^{H} q_{t,h}^w K_h^w, E^p, k^p) \tag{14}$$

Dual-Mode NAM improves THA by incorporating a single-headed phrase-mode attention network, where the phrase attention logits can be used directly for top-k phrase search:

$$indices_t^{p\_topk} = TopK(q_t^p K^p, E^p, k^p) \tag{15}$$

2.2.2. Controllable global biasing strength

Finally, a hyperparameter λ is used to adjust the global biasing strength, which allows further downscaling the contribution of the attention context to mitigate out-of-domain regressions.

$$c_t^{scaled} = \lambda c_t \tag{16}$$

The formulation is identical to the context scaler in [20]. The only difference is that our hyperparameter is applied during inference only but not at training time. Applying the scaler only at inference allows adjusting the model behavior for different ASR applications without retraining the models.

2.3. Improved synthetic context generation

We sample n-gram phrases from the target transcript and generate a synthetic context set to train the biasing module. For example, the prior works (Figure 1) [4, 10, 15] are trained with a global utterance batch size of 4096, and, at every training step, a bias phrase b_i is sampled from the transcript truth y_i to form a bias batch s_j, where each s_j is mapped to 32 utterances.

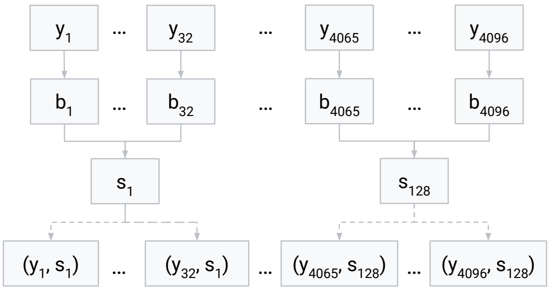

Figure 1: *Prior works: per-batch context generation.*

We further generalize by replacing the per-batch method with per-utterance context generation as follows (Figure 2):

1. Each utterance y_i can be associated with a unique batch of bias phrases s_i during training, where s_i consists of a true bias phrase b_i (1 – 8 words) and $bias\_batch\_size - 1$ phrases sampled from the bias pool B^{pool}. B^{pool} is a collection $\{b_i\}_{i=1}^{B}$ created every training step; by default $bias\_batch\_size = 32$, $B = 4096$. Sampling from the whole pool per training step improves training diversity; further, we can lower the training difficulty for y_i by lowering $bias\_batch\_size$, which is not possible when the bias batch for each utterance is reused across utterances, as in Figure 1.
2. A fraction (10%) of bias batches are shuffled, which mitigates ASR quality degradations when an utterance is associated with irrelevant bias phrases during inference.

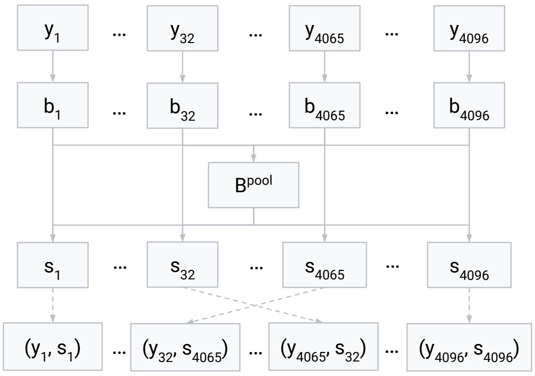

Figure 2: *Dual-Mode NAM: per-utterance context generation.*

Similar to [4], bias phrases are removed if they are prefixes of others within s_i to avoid overloading the attention network; s_i is assigned an empty batch probability of 10% to improve the ASR in no-context scenarios. For no-context scenarios (0 bias entities) during training and inference, x_t^{biased} falls back to x_t.

3. Experimental setup

3.1. Datasets

All collected experimental data sets adhere to the Privacy Principles in [21] and AI Principles in [22].

3.1.1. *Multi-domain training corpora*

The models are trained on multi-domain speech data [23] consisting of anonymized English utterances from domains including voice search, far-field and long-form. The speech transcripts contain a mix of human-transcribed labels and machine-transcribed labels produced by teacher ASR models [24].

3.1.2. *Multi-context testing corpora*

We evaluate on the multi-context TTS testing corpora as described in Section 3.2.2 of [15]. WO_PREFIX consists of 1.3K utterances chosen from $APPS, $CONTACTS, and $SONGS (denoted ACS); W_PREFIX consists of 2.6K utterances with prefixed patterns including “open $APPS”, “call $CONTACTS”, and “play $SONGS”; ANTI consists of 1K utterances that simulate general voice assistant traffic.

Each utterance can be associated with up to 3K bias entities from the ACS categories. WO_PREFIX and W_PREFIX are used to measure in-domain WERs: each utterance is assigned one transcript truth entity, with the remaining non-matching entities being distractors. ANTI is used to measure out-of-domain WER and the utterances are assigned distractors only.

3.2. Model architecture

As shown in Figure 3, the context c_t is injected into the ASR system with cascaded Conformer encoders [19, 25], which contains a 12-layer causal encoder for streaming (110.4M params), a 5-layer non-causal encoder for non-streaming (30.5M params), and a decoder network [26] (9.4M params) with a mixed-case WP vocabulary of size 4096 (embedding size is 640). All experimental ASR models are trained using the open-source Lingvo toolkit [27], on 8x8 cloud TPU v3 [28] using a global batch size of 4096. We applied the improved synthetic context training recipe in section 2.3. All parameters are randomly initialized and trained from scratch using the Adam optimizer [29]. Models were evaluated at 700K steps. For simplicity, this is a ASR-model-only evaluation which does not contain other components such as the endpointer, and in which the WERs are reported using the non-causal encoder path.

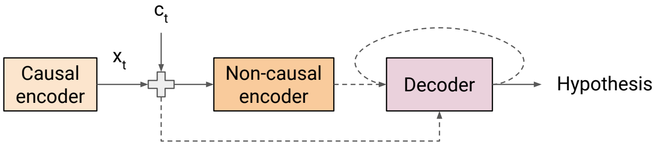

Figure 3: *ASR with cascaded Conformer encoders. We adapt the model by adding context c_t to the causal encoder output x_t.*

Model variants are listed below.

NAM The baseline ASR model that consists of:

1. A NAM WP attention network (section 2.1.2) with 855.6K params, where $Feedforward^w$ is a 2-layer RELU network with a hidden and output dimension of 256; $d = 256$, $H = 4$, $d_k^w = 128$, $d_v^w = 128$, $d_o = 512$ (same as x_t).
2. A 3-layer Transformer [30] context encoder with 3.42M params, of model dimension 256 and hidden dimension 1024. The context encoder is trained without using cls.

DUAL Dual-Mode NAM. Compared to NAM, the context encoder is trained with cls; a phrase attention network is added (1.84M params), where $Feedforward^p$ has identical settings to $Feedforward^w$; $d = 256$, $d_k^p = 128$, $d_v^p = 2048$. The attention context sampling rates are $[c_t^p, c_t^w] = [0.2, 0.8]$.

Inference options are listed below, where N depends on the number of input bias entities and L is set to 16 to cover 99.7% of the bias entities without causing truncation.

- **P** Inference using only phrase-mode attention. The runtime complexity is $O(N)$ per frame.
- **W** Inference with wordpiece-mode attention. The runtime complexity is $O(N \times L)$ per frame.
- **T** Inference with THA. The runtime complexity is $O(N+K^p \times L)$. NAM (T) follows equation (14). DUAL (T) follows equation (15) via the phrase attention network. K^p is set to 32 as a reasonable tradeoff between quality and latency.
- **TC** T plus controllable global biasing strength, set to $\lambda = 0.6$.

4. Results

4.1. Quality

As shown in Table 1, DUAL (T) (with a phrase attention augmented THA alone) improves average in-domain WERs over NAM (T) (a.k.a. NAM+) by up to 59.8% relative (9.2 → 3.7), but comes with an average out-of-domain WER degradation of 16.0% relative (2.5 → 2.9). However, by down-scaling the global biasing strength λ to 0.6, the final setting, DUAL (TC), can achieve the best of both worlds, improving the average in-domain WERs by up to 53.3% relative (9.2 → 4.3) and the average out-of-domain WERs by 12.0% relative (2.5 → 2.2).

Table 1: *Average multi-context WERs on NAM (T) and DUAL (T / TC), computed by averaging over five scenarios where (150, 300, 600, 1500, 3000) bias entities are provided per utterance.*

Expt	NAM	DUAL	
	T	T	TC
ANTI	2.5	2.9	**2.2**
WO_PREFIX	9.2	**3.7**	4.3
W_PREFIX	5.4	**2.4**	2.7

Table 2: *Detailed multi-context WER breakdown on NAM (W / T) and DUAL (W / T / TC), from 0 to 3000 bias entities.*

Expt	#	NAM		DUAL			
		W	T	P	W	T	TC
ANTI	0	**1.8**	**1.8**	2.2	2.2	2.2	2.2
	150	2.3	2.2	2.4	2.4	2.2	**2.0**
	300	2.5	2.5	2.5	2.3	2.3	**2.1**
	600	3.1	2.7	2.5	3.2	3.3	**2.3**
	1500	3.0	2.6	2.9	3.4	3.5	**2.2**
	3000	3.0	2.5	3.0	3.2	3.5	**2.4**
WO_PREFIX	0	**21.9**	**21.9**	22.9	22.9	22.9	22.9
	150	**2.4**	2.8	4.4	2.6	2.8	3.0
	300	3.1	4.8	4.8	3.0	**2.8**	3.4
	600	3.5	7.3	5.1	**3.4**	**3.4**	4.0
	1500	5.0	13.1	6.8	4.4	**4.3**	5.0
	3000	6.2	17.8	8.5	**5.0**	5.1	5.8
W_PREFIX	0	**10.5**	**10.5**	11.4	11.4	11.4	11.4
	150	2.0	2.1	2.8	**1.7**	**1.7**	2.1
	300	2.1	3.0	3.1	**2.0**	**2.0**	2.3
	600	2.8	4.6	3.6	**2.3**	**2.3**	2.4
	1500	3.6	7.4	4.4	2.8	**2.7**	3.0
	3000	4.4	10.1	4.9	3.5	**3.3**	3.8

The detailed WER breakdown can be seen in Table 2. Notably, DUAL (T) improves the in-domain WERs of the most difficult scenario (3000 bias entities) by 71.3% (17.8 → 5.1) relative on WO_PREFIX, 67.3% (10.1 → 3.3) relative on W_PREFIX; the out-of-domain WER regressions from NAM (T) to DUAL (T) can be reversed by DUAL (TC), which performs better in all out-of-domain scenarios (150 to 3000 bias entities). Although dual-mode training results in a WER increase (1.8 → 2.2) on the no-context scenario for the general traffic, it is tolerable given that the WER is already very low.

4.2. Effects of hyper-parameters k^p and λ

Table 3 shows that the in-domain WERs are not sensitive to the choice of k^p between 4 and 64, except for degrading at $k^p = 1$. The out-of-domain WER, however, monotonically decreases with increasing k^p.

Table 3: *Effect of adjusting k^p for DUAL (T), shown in the average multi-context WERs (excluding # bias entities = 0).*

k^p	1	4	8	16	32	64
ANTI	3.8	3.3	3.2	3.1	2.9	**2.8**
WO_PREFIX	4.6	**3.7**	**3.7**	**3.7**	**3.7**	**3.7**
W_PREFIX	2.8	**2.4**	**2.4**	**2.4**	**2.4**	**2.4**

Table 4 demonstrates that the out-of-domain WERs regressions can be mitigated by down-scaling the biasing strength λ, at the cost of increased in-domain WERs.

Table 4: *Effect of adjusting λ for DUAL (TC), shown in the average multi-context WERs (excluding # bias entities = 0).*

λ	1.0	0.9	0.8	0.7	0.6	0.5
ANTI	2.9	2.7	2.5	2.4	2.2	**2.1**
WO_PREFIX	**3.7**	3.8	3.8	4.0	4.3	4.9
W_PREFIX	**2.4**	**2.4**	**2.4**	2.5	2.7	2.9

5. Limitations

When p is not zero or one, sampling-based training (section 2.1.4) trains the individual phrase- and wordpiece-mode attention networks on fewer samples than they would be trained on without sampling. In our experiments, the phrase attention network is sampled 20% of the time during training, which results in significantly less training data exposure compared to the WP attention network.

In principle, we can mitigate anti-context losses by disabling biasing (i.e., setting c_t to zero) when the no-bias token's logit dominates either phrase- or wordpiece-level attention. In our experiments, we found that incorporating the no-bias token did improve the overall WERs, but it was not a clear enough signal to gate biasing on; this is a subject for future work.

6. Conclusion

We presented Dual-Mode NAM, a model architecture that can significantly improve in-domain large scale biasing WERs by following an accurate top-k phrase search process with wordpiece context injection. Dual-Mode NAM also has an effective controllable global biasing strength mechanism to mitigate out-of-domain WER regressions without retraining the model.

7. References

[1] K. C. Sim, A. Chandorkar, F. Gao, M. Chua, T. Munkhdalai, and F. Beaufays, "Robust continuous on-device personalization for automatic speech recognition." in *Interspeech*, 2021, pp. 1284–1288.

[2] G. Pundak, T. Munkhdalai, and K. C. Sim, "On-the-fly asr corrections with audio exemplars," *Proc. Interspeech 2022*, pp. 3148–3152, 2022.

[3] D. M. Chan, S. Ghosh, A. Rastrow, and B. Hoffmeister, "Using external off-policy speech-to-text mappings in contextual end-to-end automated speech recognition," *arXiv preprint arXiv:2301.02736*, 2023.

[4] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep context: end-to-end contextual speech recognition," in *2018 IEEE spoken language technology workshop (SLT)*. IEEE, 2018, pp. 418–425.

[5] M. Jain, G. Keren, J. Mahadeokar, G. Zweig, F. Metze, and Y. Saraf, "Contextual rnn-t for open domain asr," in *Proc. Interspeech 2020*, 2020, pp. 11–15. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-2986

[6] K. M. Sathyendra, T. Muniyappa, F.-J. Chang, J. Liu, J. Su, G. P. Strimel, A. Mouchtaris, and S. Kunzmann, "Contextual adapters for personalized speech recognition in neural transducers," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8537–8541.

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

[8] P. Aleksic, M. Ghodsi, A. Michaely, C. Allauzen, K. Hall, B. Roark, D. Rybach, and P. Moreno, "Bringing contextual information to google speech recognition," in *Interspeech 2015, International Speech Communications Association*, 2015.

[9] I. McGraw, R. Prabhavalkar, R. Alvarez, M. G. Arenas, K. Rao, D. Rybach, O. Alsharif, H. Sak, A. Gruenstein, F. Beaufays *et al.*, "Personalized speech recognition on mobile devices," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5955–5959.

[10] 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 *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6632–6636.

[11] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," *arXiv preprint arXiv:2006.04768*, 2020.

[12] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang *et al.*, "Big bird: Transformers for longer sequences," *Advances in neural information processing systems*, vol. 33, pp. 17 283–17 297, 2020.

[13] I. Schlag, K. Irie, and J. Schmidhuber, "Linear transformers are secretly fast weight programmers," in *International Conference on Machine Learning*. PMLR, 2021, pp. 9355–9366.

[14] M. Han, L. Dong, Z. Liang, M. Cai, S. Zhou, Z. Ma, and B. Xu, "Improving end-to-end contextual speech recognition with fine-grained contextual knowledge selection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8532–8536.

[15] T. Munkhdalai, Z. Wu, G. Pundak, K. C. Sim, J. Li, P. Rondon, and T. N. Sainath, "Nam+: Towards scalable end-to-end contextual biasing for adaptive asr," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 190–196.

[16] M. Han, L. Dong, Z. Liang, M. Cai, S. Zhou, Z. Ma, and B. Xu, "Improving end-to-end contextual speech recognition with fine-grained contextual knowledge selection," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8532–8536.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[18] K. M. Sathyendra, T. Muniyappa, F.-J. Chang, J. Liu, J. Su, G. P. Strimel, A. Mouchtaris, and S. Kunzmann, "Contextual adapters for personalized speech recognition in neural transducers," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8537–8541.

[19] A. Narayanan, T. N. Sainath, R. Pang, J. Yu, C.-C. Chiu, R. Prabhavalkar, E. Variani, and T. Strohman, "Cascaded encoders for unifying streaming and non-streaming asr," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5629–5633.

[20] M. K. Baskar, L. Burget, S. Watanabe, R. F. Astudillo, and J. H. Černocký, "Eat: Enhanced asr-tts for self-supervised speech recognition," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6753–6757.

[21] "Google's privacy principles," https://googleblog.blogspot.com/2010/01/googles-privacy-principles.html, accessed: 2023-03-01.

[22] "Artificial intelligence at Google: Our principles," https://ai.google/principles, accessed: 2023-03-01.

[23] A. Narayanan, A. Misra, K. C. Sim, G. Pundak, A. Tripathi, M. Elfeky, P. Haghani, T. Strohman, and M. Bacchiani, "Toward domain-invariant speech recognition via large scale training," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 441–447.

[24] D. Hwang, K. C. Sim, Z. Huo, and T. Strohman, "Pseudo Label Is Better Than Human Label," in *Proc. Interspeech 2022*, 2022, pp. 1421–1425.

[25] A. Gulati, J. Qin, C.-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 2020*, 2020, pp. 5036–5040.

[26] R. Botros, T. N. Sainath, R. David, E. Guzman, W. Li, and Y. He, "Tied & reduced rnn-t decoder," in *Proc. Interspeech 2021*, 2021, pp. 4563–4567.

[27] J. Shen, P. Nguyen, Y. Wu, Z. Chen, M. X. Chen, J. Ye, A. Kannan, T. N. Sainath, Y. Cao, C.-C. Chiu *et al.*, "Lingvo: a modular and scalable framework for sequence-to-sequence modeling," *arXiv preprint arXiv:1902.08295*, 2019.

[28] N. P. Jouppi, D. H. Yoon, G. Kurian, S. Li, N. Patil, J. Laudon, C. Young, and D. Patterson, "A domain-specific supercomputer for training deep neural networks," *Commun. ACM*, vol. 63, no. 7, p. 67–78, Jun 2020. [Online]. Available: https://doi.org/10.1145/3360307

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

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