Retrieve and Copy: Scaling ASR Personalization to Large Catalogs

Sai Muralidhar Jayanthi[*]
AWS AI Labs
saimucja@amazon.com

Devang Kulshreshtha[*]
AWS AI Labs
kulshrde@amazon.com

Saket Dingliwal[*]
AWS AI Labs
skdin@amazon.com

Srikanth Ronanki
AWS AI Labs
ronanks@amazon.com

Sravan Bodapati
AWS AI Labs
sravanb@amazon.com

Abstract

Personalization of automatic speech recognition (ASR) models is a widely studied topic because of its many practical applications. Most recently, attention-based contextual biasing techniques are used to improve the recognition of rare words and/or domain specific entities. However, due to performance constraints, the biasing is often limited to a few thousand entities, restricting real-world usability. To address this, we first propose a "Retrieve and Copy" mechanism to improve latency while retaining the accuracy even when scaled to a large catalog. We also propose a training strategy to overcome the degradation in recall at such scale due to an increased number of confusing entities. Overall, our approach achieves up to 6% more Word Error Rate reduction (WERR) and 3.6% absolute improvement in F1 when compared to a strong baseline. Our method also allows for large catalog sizes of up to 20K without significantly affecting WER and F1-scores, while achieving at least 20% inference speedup per acoustic frame.

1 Introduction

End-to-end ASR models based on Connectionist Temporal Classification (CTC) (Graves et al., 2006) and Transducers (Graves, 2012) are widely popular. Although these models have shown outstanding improvements over hybrid models, they often struggle to recognize uncommon domain-specific words. This is further exacerbated for streaming ASR models due to limited audio context (Chiu et al., 2021). To tackle this problem, attention-based Contextual Adapters (CA) have been proposed to boost a list of custom entity words (called 'catalog') and have showcased to work well with catalogs up to hundreds of catalog items (Sathyendra et al., 2022; Dingliwal et al., 2023). However, many industrial applications have larger

catalogs that can comprise of tens of thousands of words for ASR personalization. For example, a catalog of products sold by a business, a list of customer names or a search in video-on-demand platforms. In this work, we identify two main challenges in scaling the existing methods to larger lists: (1) Computing attention scores for each catalog item can significantly increase the latency of the system (and often redundant!), which is prohibitively critical for any streaming application, (2) Large catalogs have more phonetically similar words which makes it hard for the CA models to disambiguate the correct entity for boosting.

To address these challenges, we propose novel inference and training strategies. Through our inference method called "Retrieve and Copy", we first retrieve a smaller subset of relevant entities using Approximate Nearest Neighbor (ANN) search from the large catalog and then use only the retrieved entities for contextual biasing. Our best model leverages Fast AI Similarity Search (FAISS) (Johnson et al., 2019) that is designed for fast retrieval at scale. Further, we introduce a fine-tuning strategy using hard negatives for the CA models. We use clustering to identify phonetically similar words from the training data and help the model learn to disambiguate between them. Overall, the contributions of our work are:

- We propose "Retrieve and Copy" inference strategy for ASR personalization with large catalogs that achieves atleast 20% inference speedup per acoustic frame without affecting accuracy for large catalogs.

- We propose a fine-tuning strategy for Contextual Adapters to better disambiguate between similar sounding custom entities to improve accuracy.

- Using different datasets and catalog types, we show that our proposed methods can scale upto 20K catalog items, resulting in up-to a

---

[*]Equal Contribution

6% more WERR and 3.6% improvement in absolute F1 compared to a strong baseline.

2 Related Work

Attention-based contextual biasing modules have widely been used by ASR systems to personalize towards a catalog of a few hundred custom entities (Pundak et al., 2018; Bruguier et al., 2019; Sathyendra et al., 2022; Dingliwal et al., 2023; Munkhdalai et al., 2022). However, Munkhdalai et al. (2023) showed that inference latency increases significantly even with a few thousand catalog items. Similar to our approach, they propose to filter a small set of entities using maximum inner product. However, their method reduces dependency of phrase-length in the attention computation for associative memory based biasing modules (Munkhdalai et al., 2022). In contrast, we use a single vector to represent an entity and hence do not have this dependency. Also, their experiments are limited to catalogs of size 3K, while we scale to 20K custom entities because of retrieval methods like FAISS. Further, we introduce a fine-tuning strategy that specifically tackles the challenges of large catalog size on accuracy. Alon et al. (2019) previously used difficult examples for ASR contextualization but their methods relied on generating fuzzy alternatives using phonetic similarity metric, while Bleeker et al. (2023) used an ANN search with audio features. On the other hand, we use a simple clustering based strategy that allows us to easily use the elements belonging to the same cluster as phonetically similar entities.

3 Background

A CTC encoder takes in an audio, passes it through multiple Conformer blocks (Gulati et al., 2020), and generates a sequence of word piece posteriors. Contextual Adapters (CA) is a separate module that is added to the CTC encoder for boosting custom entities for personalization. Let $X^{1:T}$ denote T output audio feature vectors from the CTC encoder. Let $W^{1:N}$ be a list of N custom entity words. CA comprises of two main components: (i) Catalog Encoder: an LSTM (Hochreiter and Schmidhuber, 1997) that encodes word-piece sequences of custom entities into vectors (denoted by $C^{1:N}$) (ii) Biasing Adapter: an attention module (Vaswani et al., 2017) that uses X^t for each audio frame $t \in [1, T]$ as query and $C^{1:N}$ as keys to generate biasing vectors B^t. B^t is then added back to

X^t, thereby boosting any relevant custom entity. Let $\theta^Q, \theta^K, \theta^V$ represent the query, key and value matrices of the Biasing Adapter respectively. Then for each time frame $t \in [1, T]$, attention operation is equivalent to finding a score of each entity word $W^n, n \in [1, N]$ using the inner product $s_t^n = \langle \theta^Q X^t, \theta^K C^n \rangle$ and then $B^t = \sum_1^N s_t^n \theta^V C^n$.

For training this module, each audio-text pair (x, y) is augmented with a list of boosting words $W = \{x, W'\}$, wherein the word w is from the ground truth transcript y that has the least term-frequency in the entire training data and W' is a random subset of other low term-frequency words present in the training data but not in y. In this way, CA learns to distinguish word w from the rest of words W' and boost its probability in the output sequence. We choose words with lower term-frequency as they are the hardest ones to be recognized by the un-adapted CTC encoder model.

4 Methodology

In many practical applications, the number of custom entities at inference time (N) can be substantially large and can contain up to 20K entities. As highlighted in Section §1, this creates challenges for both inference speed and performance. Following are our proposed inference- and training-side strategies designed to tackle these challenges respectively.

4.1 Retrieve and Copy (RAC) Inference

In order to reduce the inference latency, we need to find efficient ways to selectively reduce the catalog size to a smaller number at inference time. For this, we propose "Retrieve and Copy", where we first retrieve the most relevant entities for a given audio and then use them for CA. Assuming either one or none of the custom entities will be spoken in a given audio, the score s_t^n of all but one would be close to 0. Therefore, the biasing vector can be approximated using $B^t = \sum_1^k s_t^n \theta^V C^n$, where $k \ll N$ and $C^{1:k}$ are the vectors of the top-k entities with the maximum inner product with the query vector ($\theta^Q X^t$) at any given time frame. This selection of top-k entities reduces linear dependence of N in the computation of attention in Contextual Adapters to k. We try different approaches for the retrieval of entities as summarized below.

Clustering: We reduce the number of entities for biasing as follows: (i) cluster entities with
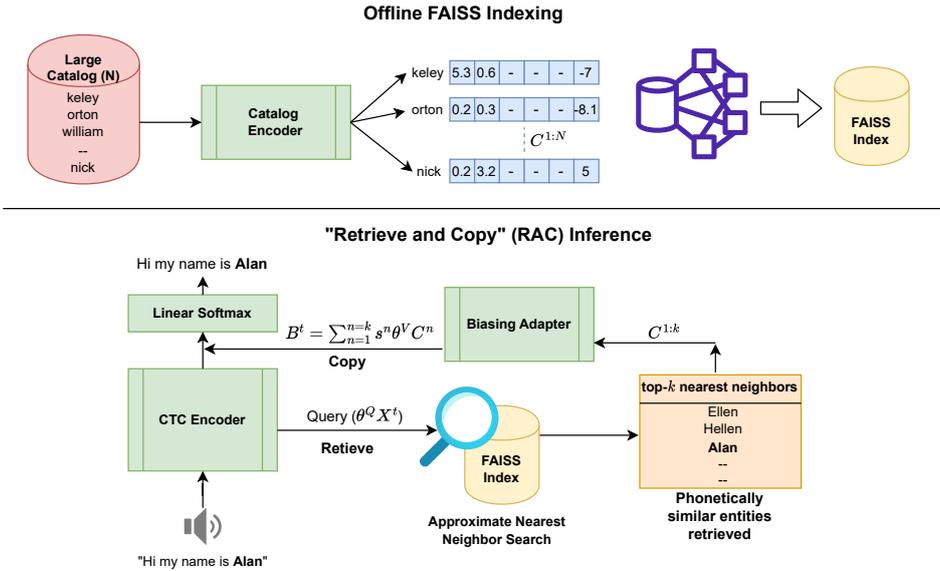
Figure 1: Details of our "Retrieve and Copy" inference strategy. (Top) Offline creation of FAISS index for a large catalog. (Bottom) Using ANN search to retrieve a subset of entities from the FAISS index for a given audio

similar vector representations, (ii) choose most relevant cluster(s) for a given audio, (iii) use only the entities in the chosen cluster(s) for biasing. For the first step, we use k-means clustering on the vectors $\theta^K C^n$ using Euclidean distance to cluster N entities into M clusters ($M < N$) offline. During inference, we score each cluster by computing the distance between the query vector at each time frame and the centroid of the cluster. Finally, we collect all the entities in each of the top-l clusters and use them for biasing with Contextual Adapters.

Approximate Nearest Neighbors (ANN): In this approach, we leverage Theorem 1 in Bachrach et al. (2014) to transform the problem of finding top-k entity vectors with maximum inner product to an ANN search problem. We transform our vectors from d-dimension to $d + 1$ and find top-k entities with the least Euclidean distance with the query vector at each time frame. Various methods have been proposed for solving ANN including FAISS and FAISS-IVF (Johnson et al., 2019), and HNSWLIB (Malkov and Yashunin, 2018).[*] As shown in Figure 1, we create an index of our transformed custom entity vectors offline such that it can be efficiently queried for top-k nearest neighbors during inference. At inference, for a given audio, we use the audio frame vector at

each time step as the query, collect top-k nearest neighbors, and then pass them to the Contextual Adapters for biasing.

4.2 Hard Negative Fine-tuning (HNFT)

As the size of the catalog increases, we can find more phonetically similar entities within it, which makes it challenging for the Contextual Adapters to accurately disambiguate the correct entity. Further, when our RAC inference strategy is applied, the set of top-k retrieved entities, used for biasing are actually nearest neighbors in the Euclidean space. On the other hand, w and W' used during training are unrelated. This creates a mismatch between training and inference with large catalogs.

In this work, we propose an additional fine-tuning stage for the Contextual Adapters to train them with similar-sounding words. We take all the low term-frequency words from CA training data (Section §3), pass them through the already trained CA Catalog Encoder and do k-means clustering into s clusters. Then during fine-tuning, for an audio-text pair with the low term-frequency word w, instead of choosing a random subset of words W', we choose words from the same cluster as w as hard negatives. In this way, model learns to disambiguate between similar sounding words. Table 1 showcases some training words belonging to the same cluster.

---

[*]Tree based ANN methods such KDTree and BallTree (Pedregosa et al., 2011), and ANNOY (ann) haven't shown practical gains in inference latency and hence their results are excluded in Table 3

ID	Training words in the same cluster
98	bowl's, bolt's, bolz, bolles, bowell's, boby, boaby
112	froing, froning, refrying, refering, furloughing
234	quake-hit, well-knit, top-knot, k-cup, pay-cuts, pay-cut
999	conjoining, congenial, conjugal, convivial, conjuncture

Table 1: Words from randomly picked clusters for HNFT

Dataset	Num Audios	Avg. Audio Length (s)	Num Words	Catalog Size	Ground Truth Entities Size
First Names	250	4.9	818	20K	250
First Names w/o CP	250	4.3	250	20K	250
Last Names	250	5.2	821	20K	250
Last Names w/o CP	250	4.7	250	20K	250
Occupation	2160	4.9	19814	9K	144
Voxpopuli	1842	9.6	44830	20K	156

Table 2: Statistics of different evaluation datasets

5 Experimental Setup

5.1 Evaluation Datasets

We extensively test our approach on five in-house conversational datasets and a public dataset. The details of the datasets are provided below and summarized in the Table 2.

First Names & Last Names: Each utterance of this dataset contains a speaker telling their first name or last name respectively. In addition, they consist of a carrier phrase (CP) such as "my name is", "my first is", "yeah it is", etc. along with the name. We use a list of 20K common first and last names as catalog for these datasets respectively. We create random subsets from the large catalog consisting of all ground truth entities to carry out experiments related to varying catalog sizes.

First Names w/o CP & Last Names w/o CP: These datasets are similar to First Names and Last names except they do not contain any carrier phrases. Again, we individually use a list of 20K names for each of these datasets.

Occupation: Each utterance of this dataset contains a speaker telling about their occupation such as "cinematographer", "mammographer", etc. They may use a long or a short carrier phrase along with the occupation typical of a conversational setting. We use a list of 9K common occupations as catalog for this dataset.

VoxPopuli (Wang et al., 2021) is a public dataset of European Parliament event recordings from which we use the English test partition for our experiments. This dataset contains long audios and the entity words compose a very small percentage of total words in the dataset. For this dataset, we create an in-house catalog consisting of first, last, city and country names as well as 92 rarest words in training split of Voxpopuli as measured against training data's term frequencies.

5.2 Evaluation Metrics

We report Word Error Rate Reduction (WERR) (%) on entire dataset and F1 scores (%) of the ground truth entities as evaluation metrics. Ground

truth entities are those that are present in both large catalog as well as test set transcripts. When computing inference latencies, we compute the wall clock time overhead of the contextual adapters module attached to the streaming ASR (Section §3) model in milliseconds (ms) per audio on a single CPU machine without multi-processing.

5.3 Models

We evaluate three models – streaming ASR model without contextual adapters (**Baseline**) and with contextual adapters (**CA**), and a model further tuned with our proposed finetuning strategy in Section §4.2 (**HNFT**).

Our models are trained with ESPnet (Watanabe et al., 2018) using Conformer blocks (Gulati et al., 2020) and joint CTC-Attention framework (Kim et al., 2017; Watanabe et al., 2017) with Adam optimizer (Kingma and Ba, 2014). The Baseline model is trained with 50K+ hours of speech-text parallel corpus in English. For CA & HNFT models, we follow Dingliwal et al. (2023)'s proposal and curate a subset of 1K hour from the parallel corpus leading to 230K catalog entities for adapting. For HNFT, we cluster these 230K catalog based on their embeddings from CA model into $s = 1000$ clusters. For inference, we finalize the RAC hyper-parameters (k, M, l) based on the performance of First Names dataset. Further, we train a 4-gram language model (LM) using the parallel corpus's texts for shallow fusion (Kannan et al., 2018) during ASR decoding. We refer the reader to Appendix A for more details on model training and implementation.

6 Results

RAC Inference achieves the lowest inference latency with no performance regression: In Table 3, we compare the inference latencies and F1 scores in the retrieval of entity words on two of our datasets. We compare the Baseline model, CA with and without our RAC inference strategy, providing a comparison of different retrieval methods. First,
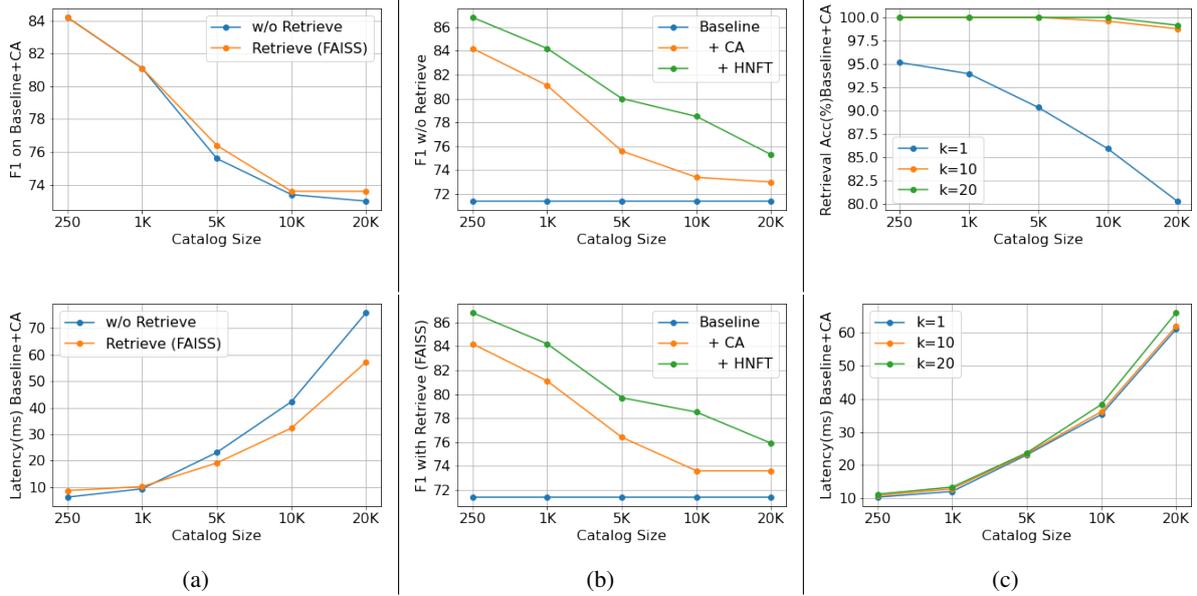
Figure 2: **(a)** Effect of RAC inference on F1 (top) and Latency (bottom) for varying catalog sizes. **(b)** Effect of HNFT on F1 without (top) and with RAC (bottom) for varying catalog sizes. **(c)** Effect of hyper-parameter k in FAISS on retrieval accuracy (top) and latency (bottom) for varying catalog sizes.

	Methodology	First Names		Last Names	
		F1	Lat (ms)	F1	Lat (ms)
1	Baseline	71.4	N/A	67.9	N/A
2	+ CA	73.0	75.9	74.8	84.7
3	+ Clustering	73.0	71.6	73.5	64.8
4	+ HNSWLIB	73.6	62.2	75.4	57.7
5	+ FAISS-IVF	72.8	**50.6**	73.6	**52.3**
6	+ FAISS	**73.6**	60.7	**75.4**	57.5

Table 3: Performance comparison of Baseline, CA and different retrieval methods of our proposed RAC inference

we observe that our proposed inference strategy yields significant improvements in latency over the standard inference with large catalog. Particularly, our best retrieval method FAISS reduces latency by 20-32% (75.9ms to 60.7ms and 84.7ms to 57.5ms) compared to the inference without retrieval (Row 2 v/s 6). Second, we see that this speedup doesn't come at the cost of performance, as our FAISS-based method achieves the best F1 score for both the datasets. This means the entities retrieved by FAISS based ANN search almost always contain the correct entity. We confirm this hypothesis later in our experiments. In fact, we observe some minor improvements in performance due to possible removal of unrelated entities in the retrieval step. Finally, among all the retrieval methods chosen for our experiment, FAISS ($k = 10$) performs the best. On the other hand, Clustering ($M = 2000, l = 4$) performs worse than ANN based retrieval methods

(Row 3) in both performance and latency. Due to it's performance, FAISS will be our choice for retrieval in all the subsequent experiments. We further validate our claim of retaining the performance (or even improving) when we use our proposed efficient inference strategy on more datasets in the Table 4 (Row 2 v/s 4, Row 3 v/s 5).

HNFT improves the WERR/F1 of Contextual Adapters: In Table 4, we present an extensive comparison of our proposed training approach against baseline on different datasets. Comparing F1 scores (Row 2 v/s 3, Row 4 v/s 5), we note that our fine-tuning strategy using hard negatives outperforms CA in biasing the correct custom entity for all the datasets. The WERR (%) also improves or remains similar for all the datasets. For Voxpopuli, while the F1 score improves significantly, there are WERR regressions with the use of CA with a large catalog. This is because the custom words are only $< 0.2\%$ of total number of words in the dataset. While CA can recognize the right entity word (which are typically the most important words of the utterance), they sometimes unnecessarily substitute common words. This results in an increase in overall WER, which is in line with previous findings on the use of contextual biasing (Munkhdalai et al., 2023). In Table 5, we show qualitative examples of the output of our models on First Names dataset. The CA boosts either a phonetically similar word (*Ruben*)

	Model	First Names		Last Names		First Names w/o CP		Last Names w/o CP		Occupation		Voxpopuli	
		WERR (%)	F1(%)	WERR (%)	F1(%)	WERR(%)	F1(%)	WERR(%)	F1(%)	WERR(%)	F1(%)	WERR(%)	F1(%)
	w/o Retrieve												
1	Baseline	0.0	71.4	0.0	67.9	0.0	68.6	0.0	54.7	0.0	54.8	**0.0**	65.5
2	+ CA	1.1	73.0	13.1	74.8	4.2	70.1	10.0	60.0	**0.4**	63.2	-2.6	69.5
3	+ HNFT	5.3	75.3	13.0	77.5	7.3	71.6	15.0	62.5	0.2	**65.2**	-5.5	**71.5**
	Retrieve (FAISS)												
4	+ CA	1.6	73.6	**14.0**	75.4	5.8	70.8	10.0	60.0	0.1	63.5	-2.7	70.3
5	+ HNFT	**5.9**	**75.9**	11.6	**77.8**	**8.1**	71.9	**16.0**	**63.6**	-1.2	65.1	-5.6	**71.5**

Table 4: WERR (%) and F1 (%) scores for models described in Section 5.3 with and w/o retrival based inference. The WER of our Baseline model for VoxPopuli dataset is 10.5, in line with streaming models of similar sizes.

Model	Transcription
Baseline	my name is Ruben
+ CA	my name is Ruben
+ HNFT	my name is Rueben
Baseline	my name is Wally
+ CA	yes it's Wy
+ HNFT	yes it's Wally

Table 5: Examples of generated transcripts for various models described in Section 5.3 on First Names dataset.

or struggle to boost any entity word (*Wy*) from the catalog. However, once we train our model with phonetically similar entities, it can disambiguate the subtle difference between these entities and can recognize the correct entity (*Rueben* and *Wally*).

In Figure 2, we study the effect of varying catalog size on different latency and accuracy metrics for our proposed methods. We use First Names dataset for the ablation and randomly subsample our total 20K catalog into subsets of smaller sizes (250, 1K, 5K and 10K). For each subset, all the ground truth entity words are retained in the subset so that we can independently study the impact of the size of the catalog.

The latency improvements of RAC Inference over baseline increases with increase in catalog size: In the Figure 2a, we compare the F1 and latency of our CA model with and without FAISS retrieval. We observe that there is a consistent decrease in F1 and an increase in latency as the catalog size increases (blue line). This validates our identified challenges of the problem of scaling contextual biasing methods to large catalogs. We also observe that our proposed FAISS based inference can help reduce the increase in latency with the increase in catalog size while maintaining similar F1 scores (orange line). Notably, the difference in the latency between our inference strategy and baseline increases with the increase in catalog size. This suggests that we can possibly go beyond catalogs of size 20K without a lot of increase in latency with our method.

HNFT improves accuracy consistently for all catalog sizes: In Figure 2b, we compare the F1 scores of our proposed fine-tuning with hard negatives strategy to Baseline and CA for different catalog sizes. We observe that the improvements of our method in the retrieval of the correct entity are consistently equal for different catalog sizes (orange vs green line). This holds for both with and without our RAC inference strategy (bottom and top sub-figures respectively). While our training method definitely improves the F1 scores, the benefits of using contextual biasing approach diminish over the Baseline for very large catalog sizes (blue line).

top-k ANN search can retrieve the correct entity with almost 100% accuracy: Finally, we try to understand the trade-off between latency and performance for different choices of our hyper-parameter k in the Figure 2c. In the top sub-figure, we plot retrieval accuracy, defined as the percentage of audios in which the correct entity was retrieved by top-k ANN search using FAISS. Notably, for $k = 1$, the accuracy drops significantly with large catalog size but we observe a very high retrieval accuracy (close to 100%) for $k = 10$ indicating that we do not lose any important entity as we select a subset of the large catalog to copy. In the bottom sub-figure, we observe that the latency is similar for different k. Hence, we choose $k = 10$ as an optimum value for all our experiments.

7 Conclusions

In this work, we identified the challenges with the use of contextual biasing methods for an industrial use case of biasing towards large catalogs. As the size of the catalog increases, we see a significant increase in latency and a corresponding drop in accuracy, making it practically infeasible to use existing approaches at such scale. To mitigate these challenges, we propose a "Retrieve and Copy" inference that leverages efficient ANN

search methods like FAISS to selectively choose a small subset of relevant entities per audio, thereby improving inference latency by at least 20%. Additionally, to improve the accuracy, we propose a fine-tuning strategy that uses phonetically similar words as hard negatives to train the model. It yields up to 6% more WER reduction and up to 3.6% absolute increase in F1 scores on one of our datasets.

Ethics Statement

We hereby acknowledge that all of the co-authors of this work are aware of the provided ACL Code of Ethics and honor the code of conduct. In this work, we focus on scaling personalization of ASR systems to large catalog lists using contextual biasing modules. For our experiments, we use a Baseline model trained using 50K+ hours of a large paired audio-text English data. Though large, we do not claim that this data is representative of all groups, accents and use cases. Our biasing mechanism can be effective in bridging the gap in the performance disparity for different groups by allowing for large custom lists. However, our methods and models are still susceptible to generating better outputs for certain groups of users. For example - even a 20K list of first names might miss names from particular communities more than the others. Therefore, scaling beyond 20K entities might be necessary to make our method inclusive of a range of users and will be studied as part of future work.

Limitations

Our RAC inference methodology improved the latency in scaling contextual biasing for large catalogs but we still see a consistent drop in F1 with increasing catalog size (2b). Incorporating hard negatives based fine-tuning helped, but more work is needed to scale our approach to even larger catalog size. Secondly, contextual biasing approaches can help in biasing the relevant entity but they can cause regressions on other common words in the dataset. In our experiments, we found that using CA on datasets with long audios like VoxPopuli can have WER regressions, specially with catalog of large size. This is another challenge in scaling these systems to some practical use cases that we plan to tackle in a future work. Lastly, privacy and intellectual property concerns prevent us from releasing the training and evaluation datasets, limiting replication by other researchers.

References

ANNOY library. https://github.com/spotify/annoy. Accessed: 2023-07-15.

Uri Alon, Golan Pundak, and Tara N Sainath. 2019. Contextual speech recognition with difficult negative training examples. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6440–6444. IEEE.

Yoram Bachrach, Yehuda Finkelstein, Ran Gilad-Bachrach, Liran Katzir, Noam Koenigstein, Nir Nice, and Ulrich Paquet. 2014. Speeding up the xbox recommender system using a euclidean transformation for inner-product spaces. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 257–264.

Maurits Bleeker, Pawel Swietojanski, Stefan Braun, and Xiaodan Zhuang. 2023. Approximate nearest neighbour phrase mining for contextual speech recognition. In *Interspeech*.

Antoine Bruguier, Rohit Prabhavalkar, Golan Pundak, and Tara N Sainath. 2019. Phoebe: Pronunciation-aware contextualization for end-to-end speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6171–6175. IEEE.

Chung-Cheng Chiu, Liangliang Cao, Olivier Siohan, Ruoming Pang, Thibault Doutre, and Wei Han. 2021. Bridging the gap between streaming and non-streaming automatic speechrecognition systems through distillation of an ensemble of models. In *Interspeech'2021*.

Saket Dingliwal, Monica Sunkara, Srikanth Ronanki, Jeff Farris, Katrin Kirchhoff, and Sravan Bodapati. 2023. Personalization of ctc speech recognition models. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 302–309.

Alex Graves. 2012. Sequence transduction with recurrent neural networks.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Anjuli Kannan, Yonghui Wu, Patrick Nguyen, Tara N. Sainath, ZhiJeng Chen, and Rohit Prabhavalkar. 2018. An analysis of incorporating an external language model into a sequence-to-sequence model. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5828.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In *Proc. ICASSP*, pages 4835–4839.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Jaesong Lee and Shinji Watanabe. 2021. Intermediate loss regularization for CTC-based speech recognition. In *Proc. ICASSP*, pages 6224–6228.

Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836.

Tsendsuren Munkhdalai, Khe Chai Sim, Angad Chandorkar, Fan Gao, Mason Chua, Trevor Strohman, and Françoise Beaufays. 2022. 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)*, pages 6632–6636.

Tsendsuren Munkhdalai, Zelin Wu, Golan Pundak, Khe Chai Sim, Jiayang Li, Pat Rondon, and Tara N. Sainath. 2023. Nam+: Towards scalable end-to-end contextual biasing for adaptive asr. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 190–196.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Golan Pundak, Tara N Sainath, Rohit Prabhavalkar, Anjuli Kannan, and Ding Zhao. 2018. Deep context: end-to-end contextual speech recognition. In *2018 IEEE spoken language technology workshop (SLT)*, pages 418–425. IEEE.

Kanthashree Mysore Sathyendra, Thejaswi Muniyappa, Feng-Ju Chang, Jing Liu, Jinru Su, Grant P Strimel, Athanasios Mouchtaris, and Siegfried Kunzmann. 2022. Contextual adapters for personalized speech recognition in neural transducers. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8537–8541. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. *arXiv preprint arXiv:2101.00390*.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. ESPnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech*, pages 2207–2211.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. 2017. Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253.

A   Model Training

Our models are trained with joint CTC-Attention framework (Kim et al., 2017; Watanabe et al., 2017) and intermediate CTC regularization (Lee and Watanabe, 2021) with 20 layers of Conformer blocks (Gulati et al., 2020) consisting of 8 attention heads and 512 hidden dimension. During inference, we discard the Attention head and use CTC decoder for transcript generation. We train the Baseline model with 50K+ hours of speech-text parallel corpus in English consisting of a mix of accents, speakers, sampling rates and background noise.

All our models are trained with Adam optimizer (Kingma and Ba, 2014). We train the Baseline model for 30 epochs and continue training the CA model from its last checkpoint for 50 epochs by freezing all but adapter parameters. We adopt curriculum training for the CA model by linearly increasing the biasing catalog size during training from 30 to a maximum of 200 in steps of 4 per epoch and using random negatives drawn from the pool of 230K catalog. We hypothesize that gradually expanding the catalog size can make the model more robust to large catalog settings. HNFT model is finetuned on top of the CA model for 10 epochs.[*]

We train a SentencePiece (Kudo and Richardson, 2018) tokenizer with token size of 2048 for encoding transcripts. Further, we train a 4-gram language model (LM) using the parallel corpus's texts for shallow fusion (Kannan et al., 2018). We keep the tokenizer and the LM same across all the models. During inference, we use a beam size of 50 and LM weight of 0.6 in all our experiments. Our work is implemented in the open-source toolkit ESPnet (Watanabe et al., 2018).

---

[*]Our experiments indicate that training for more than 10 epochs has no significant impact