An Efficient Self-Learning Framework For Interactive Spoken Dialog Systems

Hitesh Tulsiani [*1] **David M. Chan** [*1 2] **Shalini Ghosh** [*1]
Garima Lalwani [1] **Prabhat Pandey** [1] **Ankish Bansal** [1] **Sri Garimella** [1] **Ariya Rastrow** [1] **Björn Hoffmeister** [1]

Abstract

Dialog systems, such as voice assistants, are expected to engage with users in complex, evolving conversations. Unfortunately, traditional automatic speech recognition (ASR) systems deployed in such applications are usually trained to recognize each turn independently and lack the ability to adapt to the conversational context or incorporate user feedback. In this work, we introduce a general framework for ASR in dialog systems that can go beyond learning from single-turn utterances and learn over time how to adapt to both explicit supervision and implicit user feedback present in multi-turn conversations. We accomplish that by leveraging advances in student-teacher learning and context-aware dialog processing, and designing contrastive self-supervision approaches with Ohm, a new online hard-negative mining approach. We show that leveraging our new framework compared to traditional training leads to relative WER reductions of close to 10% in real-world dialog systems, and up to 26% on public synthetic data.

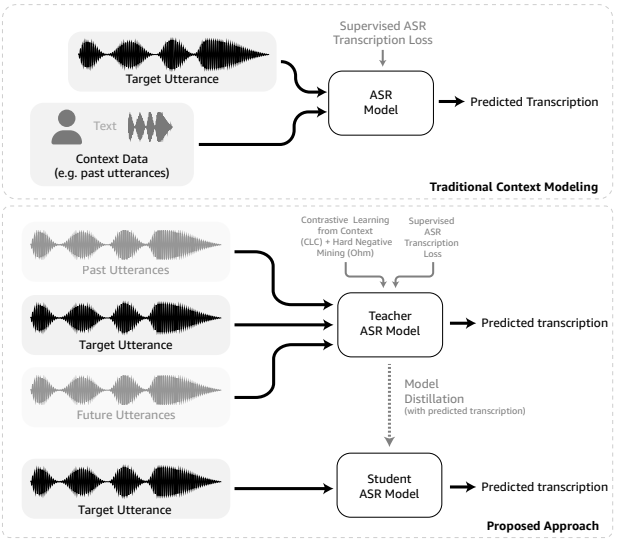

Figure 1: Traditional dialog systems learn to perform ASR using only supervised feedback with large-scale unsupervised/semi-supervised pre-training on single isolated utterances. This work introduces a novel general framework leveraging student-teacher distillation, contrastive learning, and online hard-negative mining, allowing ASR systems to learn from contextual clues and implicit feedback present in full conversational transcripts. Our two stage system naturally allows us to distill contextual signals from a context-aware teacher model to a context unaware student model.

1. Introduction

Automatic speech recognition (ASR) for dialog systems has traditionally been a focused field, where the primary goal is to produce a text transcript for an utterance given the acoustic signal corresponding to that utterance (Radford et al., 2023; Baevski et al., 2020; Hsu et al., 2021; Mitra et al., 2023). While such systems have been largely successful, particularly in the domain of dialog systems and voice assistants (leading to word error rates below 2% on the Librispeech benchmark (Radford et al., 2022)), in real-world applications such single-utterance systems have been shown to struggle with a long-tailed distribution of rare words, proper nouns, etc., leading to decreased user satisfaction with such systems (Schwarz et al., 2023; Kim & Metze, 2018; Chang et al., 2021; Chen et al., 2019; Sathyendra et al., 2022; Wei et al., 2021).

Such a struggle with long-tailed distributions has led to several promising directions of research aimed at specializing large-scale general models to handle rare words. These approaches generally center around fine-tuning where models are tuned on rare words as they are discovered (Li et al., 2022; Ku et al., 2024; Gao et al., 2022; Chang et al., 2022; Yang et al., 2023; Hung et al., 2023), "ASR model personalization" where model parameters are locally adapted with user-specific context (Gourav et al., 2021; Biadsy et al., 2022; Shor et al., 2019), or "Contextual biasing" where model inputs include additional user-specific context as part of the input to the model (Jayanthi et al., 2023; Kim & Metze, 2018; Tang et al., 2024; Sathyendra et al., 2022; Chang et al., 2021; Chen et al., 2019; Wei et al., 2021; Dingliwal et al., 2023). While these approaches have shown promising results, they often require additional compute during training or

[*]Equal contribution [1]Amazon AGI [2]UC Berkeley (work done while at Amazon). Correspondence to: Shalini Ghosh <ghoshsha@amazon.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

additional storage and retrieval for model parameter adapters, leading to significant compromises in terms of critical latency factors. These methods also often must rely on additional supervised training data during the training stage, leading to increased real-world system costs (such as data labeling) that are often not justifiable by marginal performance increases.

In this work, we aim to address the two key challenges – run-time performance and increased data costs – by introducing a two-stage framework for context-aware automatic speech recognition. To reduce run-time performance costs, we explore the use of model distillation in a student-teacher framework — we leverage context signals during training of the teacher model, but do not use context signals during run-time inference of the student model for efficiency. To reduce data costs, we leverage recent advances in self-supervised learning from intrinsic contextual signals, augmented with a novel algorithm for online hard-negative mining — this enables the teacher model to learn context signals in a self-supervised fashion, eliminating the need for additional supervised data during training.

Evaluating the real-world performance of such a system is challenging, as there is little publicly available dialog data. To evaluate our approach, we run experiments on a large dataset of over 200K hours of real-world de-identified data from a popular conversational assistant system and show that leveraging context can help to significantly improve teacher model performance.

Our key contributions are as follows:

- We introduce a multi-stage teacher model for automatic speech recognition in contextual dialog systems which is capable of leveraging both explicit context signals (through audio and text context) and implicit feedback signals (through contrastive learning combined with a novel online hard-negative mining algorithm) present in sequential task-oriented dialogues.
- We leverage our teacher model in a distillation framework, and demonstrate that context signals can be distilled into a student model requiring no additional run-time compute compared to conventional systems.
- We demonstrate close to a 10% relative WER improvement in real-world dialog systems applications for the teacher model, and up to 24.4% WERR on the public OD3 dataset. Further, we demonstrate close to a 4% relative WER improvement when our teacher model is distilled to the student model; providing strong evidence that learning from context at training time can be effective at test time (even when such context is unavailable). We additionally show our approach does extremely well in lower-resource domains, demonstrating up to a 22.8% WERR on the tail distribution of real-world data.

Terminology We refer to our system as "self-learning" to convey the system's ability to iteratively improve its performance by learning from dialogue contexts and user feedback. This goes beyond traditional SSL (self-supervised learning) techniques by integrating an "interactive" (though offline) component where the system learns from its environment rather than solely relying on pre-existing unlabeled data.

2. Background & Related Work

Methods for modeling context for automatic speech recognition (ASR) systems can be generally categorized into two main categories: supervised methods, which rely on additional data and labels to infer context which is useful for speech recognition, and unsupervised methods, which learn context cues directly from the utterance, and any associated prior/future utterances. In this section, we discuss our proposed approach in context with prior approaches for context-aware ASR.

2.1. Supervised Context Modeling

As discussed in section 1, supervised context modeling for automatic speech recognition largely falls into three categories:

- **Fine-tuning**: where models are fine-tuned on specific datasets to increase global context awareness.
- **Model Personalization**: where model *parameters* are updated on a per-user basis using a small set of user-specific samples.
- **Contextual biasing**: where models take additional context as input during the training and inference stages.

Each of these approaches has benefits and drawbacks. Perhaps the most common approach for context modeling is fine-tuning, which includes context by training on specialized datasets (Li et al., 2022; Ku et al., 2024; Gao et al., 2022; Chang et al., 2022; Yang et al., 2023; Hung et al., 2023). Such an approach can be quite effective, as it turns a long-tail distribution problem into an in-domain problem. However, it requires the collection of explicit data for the target problem, and the scope of the context that a model can learn is limited to the collected data. Further, this data collection process is often expensive – thus fine-tuning is often employed largely as an augmentation to an existing pre-trained model to fix specific errors, rather than as a good method for improving context awareness in general.

While fine-tuning adjusts the model globally to incorporate context (such as rare words), recently some approaches have been explored that focus on adjusting the model parameters locally to account for context. Gourav et al. (2021) show that small personalized models can be effective at incorporating information from user contexts, and Biadsy et al. (2022) show that small model adapters consisting of only a few

thousand parameters can be locally fine-tuned for each user to improve ASR recognition performance. Shor et al. (2019) show that such models can be trained using as little as five minutes of personalized speech. These approaches represent good ways of fine-tuning models to focus on users' individual needs, however, training model adapters for each user can be expensive and comes with storage requirements, inference performance questions, and data privacy concerns (as speech needs to be processed in the cloud, often with batches of other user data).

Instead of adjusting the model parameters, contextual biasing moves the inclusion of context to the input domain. Several types of context are effective including user information (such as contact names) (Tang et al., 2024; Sathyendra et al., 2022), prior utterances (Chang et al., 2021), visual clues (Hsu et al., 2021; Chan et al., 2022), text catalogs (Dingliwal et al., 2023; Chan et al., 2023). Outside of ASR, contextual biasing has long been shown to be effective in NLP applications (Novotney et al., 2022; Shenoy et al., 2021; Zhao et al., 2019; Liu & Lane, 2017; Jaech & Ostendorf, 2018; Kim & Metze, 2018; Lin et al., 2015; Williams et al., 2018; Munkhdalai et al., 2022; Sun et al., 2023). While contextual biasing represents an important component of context modeling, it is often limited by the requirement to collect supervised data during the training phase (i.e. contexts need to be explicitly collected and stored), as well as the requirement to have contexts during the inference phase, which can lead to significantly degraded performance in context-free scenarios. Further, contextual biasing often suffers from increased model complexity during inference, leading to slower response times and decreased user satisfaction.

2.2. Unsupervised Learning From Dialogue Contexts

Instead of learning context explicitly, unsupervised learning of context clues is a largely under-explored area in automatic speech recognition. Recent work has started to explore how we can learn contextual information from audio context alone. Hori et al. (2021) and Hori et al. (2020) take in several utterances at once, and use this joint context to perform automatic speech recognition on the final target utterance (demonstrating up to 15% improvements in WER). Unfortunately, these methods require previous utterances to be available at test time and suffer when no previous context is available. Using only the target utterance, Chan & Ghosh (2022) show that other unrelated utterances within a batch can be used to filter noise from automated speech recognition models, however, they do not show that such methods help beyond global and local noise removal.

Instead of using audio, Kim et al. (2019b) apply BERT to the partial ASR transcript generated so far and use those BERT embeddings to inform the generation of the next token (effectively fusing the language model with the speech model). Similarly, both Chang et al. (2023) and Duarte-Torres et al. (2024) show that taking in related text context from past utterances can improve ASR performance. These approaches, while interesting, focus primarily on text embeddings of prior context, and do not show that such ASR performance can persist in a context-free scenario (as is often the case in on-device learning) or discuss the inclusion of future context (available at train, but not inference time).

The approach of unsupervised learning from dialog contexts is closely inspired by Chan et al. (2023) who introduce a family of methods (CLC) for learning from both past and future dialog contexts, using contrastive learning between the latent representations of past/future dialogues and the latent representation of the target utterance. The motivation behind this work is that audio that shares similar dialog contexts should have more similar latent representations, and thus, is more likely to contain relevant acoustic information. While our current approach borrows the PF-CLC objective from Chan et al. (2023) as an additional pre-training objective on top of our fully supervised and self-supervised fine-tuning process, we also found that alone, PF-CLC led to only minor improvements in overall performance due to the small per-gpu batch sizes used during training (in our case, each GPU has a maximum batch size of 16). Thus, to improve the performance of PF-CLC in our real-world training scenario, we introduce a novel scheme for online hard-negative mining, allowing for improved efficiency when applying the CLC losses during fine-tuning. Further, Chan et al. (2023) does not study in detail how to embed contexts during training, the impact of training on both past and future contexts, or if this context training persists under distillation.

2.3. Model Distillation

Model distillation (Buciluǎ et al., 2006; Hinton et al., 2015) has long been an effective tool when used to improve the performance of models during inference time. Not only are student models often more efficient than teacher models, but surprisingly, such models are often more effective on downstream test data (Radosavovic et al., 2018; Zhang et al., 2019; Pham et al., 2022). These trends have held in ASR as well, where Huang et al. (2018); Kim et al. (2019a); Mun'im et al. (2019) all show that large teacher ASR models can be distilled to resource-efficient but performant student models.

Beyond model compression, however, model distillation has more recently also been used effectively to bridge streaming models and non-streaming models, as both Yu et al. (2021) and Kurata & Saon (2020) have shown that using distillation between model architectures can lead to overcoming fundamental architecture limitations at inference time. Recently, Futami et al. (2022) showed that ASR can be improved by distilling in language models such as BERT, however, they did so using a vector-based representation, unlike our proposed approach that leverages model distillation and self-supervision (Pham et al., 2022). Closest to our framework, Masumura

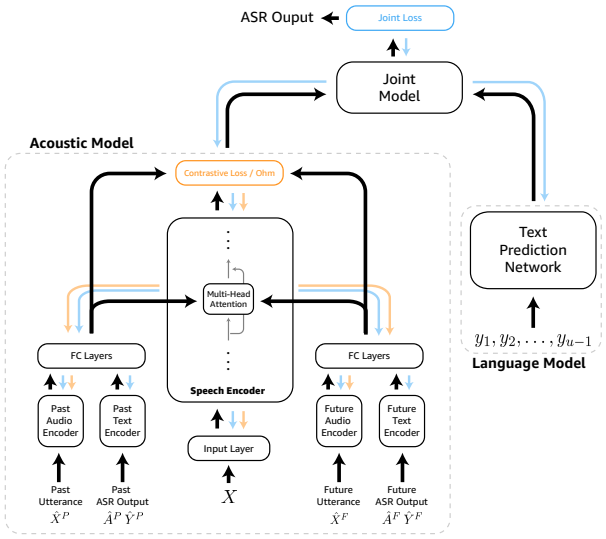

Figure 2: An overview of our approach. During training our teacher model ingests context from past/future audio and text along with the current utterance, and learns both implicitly and explicitly using CLC (Chan et al., 2024) for implicit context learning and supervised joint loss for explicit learning from supervised data. ↑ show data flow in forward-pass, and ↓ ↓ show loss propagation from each of the components.

et al. (2021) use distillation to bridge models that have long audio input contexts (such as several input utterances) to single utterance models, however their approach is limited to using text models in context, and they do not explore using audio context or learning the contextual hints from dialog.

3. Self-Learning for Dialogue ASR

An overview of our approach is given in Figure 1, and consists of two key components: a context-aware teacher model, leveraging both explicit context signals and implicit user feedback in the dialogue, and a single-utterance student model distilled from the context-aware teacher.

3.1. Teacher Model

Following best practices, our teacher model is composed of a Conformer-based transducer network (Graves, 2012) - a nonstreaming model which can attend to all frames in an utterance. In addition, our teacher model also leverages both past and future contexts as explained in the following sections.

3.1.1. Learning from Explicit Context

In this work, we leverage several explicit context sources drawn from the dialogues themselves. The first is the audio context, formed by the sequence of user input queries in a given dialogue (preceding and succeeding audio context is represented as X^P and X^F respectively in Figure 2). The second is text context, the ASR one-best hypothesis (indicated as $\hat{Y}^P$ and $\hat{Y}^F$) corresponding to the sequence of user input queries in a dialogue along with the response generated by assistant encoded in text form (indicated as A^P and A^F).

Traditionally, a transducer-based system, at each time step, outputs a probability distribution over its vocabulary (word-pieces) conditioned on the acoustic observations $X = x_1, x_2, ..., x_T$ and previously observed word-piece tokens $y_1, y_2, ..., y_{u-1}$, which could be expressed as $P(y_u|X, y_1, y_2, ..., y_{u-1})$. To model the explicit long-term context in the teacher model, we extend the above equation by further conditioning on the set of context signals, $Z = \{\hat{Y}^P, \hat{Y}^F, A^P, A^F, X^P, X^F\}$, to get $P(y_u|X, y_1, y_2, ..., y_{u-1}, Z)$.

Audio context modeling Like Hori et al. (2021) we explore two methods for adding audio context from dialogues:

Feature Concatenation: In feature concatenation, we concatenate features of past and future utterances along with the seed utterance and pass it through the audio encoder. Encoder outputs are then segmented to extract embeddings corresponding to seed utterance and are combined with prediction network output to compute transducer loss. The presence of a self-attention network in the audio encoder allows us to learn the dependency on context streams.

Audio Embeddings: For audio embeddings, past and future context is encoded via a separate encoder (called "context encoder"). We use either a HuBERT pre-trained conformer encoder or the audio encoder of the transducer network as the context encoder. These audio embeddings are passed through a multi-headed self-attentive pooling layer (Chang et al., 2023) and concatenated in time dimension with keys and values of self-attention module (MHSA) in the audio encoder (represented in Figure 2. Thus the inputs to MHSA (query - q, key - k, value - v) can be represented as $q = X; k = [X^P, X, X^F]; v = [X^P, X, X^F]$. This allows us to attend to the contextual signal on a per-query basis. Note here that the output and input of the MHSA module still have the same number of time frames. This ensures that no other component in the model needs to be modified. Another distinct advantage of re-purposing MHSA in this manner as opposed to introducing a separate cross-attention layer (to attend to context) is that it allows us to easily extend conventional single utterance models to be context aware.

Text context modeling In addition to audio context, following Duarte-Torres et al. (2024), Kim et al. (2019a), and Chang et al. (2023), we explore two variants for encoding text context from prior utterances in a dialog:

BERT Embeddings: In the BERT embedding case, we leverage a pre-trained text embedding model (Devlin et al., 2019) with 5M parameters based on BERT to generate a summary vector for the past/future text representations.

Learned Embeddings: In the learned embedding case, text context is tokenized using a sentence piece model (Kudo & Richardson, 2018) and each token is represented as a one-hot vector over vocabulary size. This is then converted to an em-

bedding and combined in the self-attention layer of the audio encoder (similar to the audio embeddings described above).

3.1.2. Learning from Implicit Context

In a multi-turn interaction with a voice assistant, the user may repeat or rephrase their query (to correct the system) following an unexpected response by the assistant. To empirically establish that such user reformulations (implicit interactions) are correlated with ASR, we conducted a simple experiment. We prepared two datasets: (i) Natural sampling: data is uniformly sampled to form our test set; and (ii) Reformulation sampling: we sample utterances that cause the user to repeat or rephrase their query. We then evaluate both our existing teacher and student models on these datasets. In this experiment, we observed that both the teacher and student models have significantly higher word error rates (11% and 15% respectively) on the reformulation sampling dataset compared to uniform sampling. This observation, combined with the fact that approximately 15% of analyzed interactions had user reformulations, shows that user-provided implicit feedback can correct ASR errors. Please note that such feedback is not directly solicited through the dialogue but inferred from user corrections and follow-up queries, hence we refer to it as "implicit".

Thus, while it is possible to learn to leverage context signals from the explicit context, it is also important to learn from implicit signals in the data. Recently, Chan et al. (2024) showed that implicit context present in the dialogues can be used to further augment the training process through contrastive learning. Drawing on their work, in this work, we leverage the past-future CLC objective from Chan et al. (2024) (which we refer to as PF-CLC), to incorporate implicit context in addition to the explicit context discussed in the previous section. In this PF-CLC approach, the positive pairs contain past/future/current utterances from the same utterance, while negative pairs are formed by past/future pairs from other utterances in the batches, further encouraging the model to organize the latent space semantically in addition to phonetically during pre-training.

3.1.3. Online Hard Negative Mining (Ohm)

When training our self-learning based system, we found a significant correlation between the local GPU batch size and the performance of the pre-training. We hypothesize that this correlation is caused by the PF-CLC learning introduced in (Chan et al., 2024), as with smaller local batch sizes, there are fewer "hard negatives" in each batch, leading to reduced efficiency when training with CLC-based losses. Unfortunately, scaling the local GPU batch size can be practically difficult, without introducing complex optimizers and training procedures for model sharding. This presents a challenging issue: how can we train contrastive models under restricted local batch sizes?

While several technical methods have been developed for contrastive learning with small batch sizes including BASIC (Pham et al., 2023), which leverage tools from gradient checkpoint and model parallelism to improve the "effective" batch size, such methods have significant compute bottlenecks, and still rely on some form of all-reduce to compute the global contrastive loss. These all-reduces are technically complex, and on many GPU clusters can lead to significant communication overhead when machines must communicate with non-local devices.

Instead, of such a complex all-reduce based approach, we target the root of the problem by aiming to build more effective local batches (i.e. batches that will induce high contrastive loss by leveraging hard negative mining, introduced by Robinson et al. (2021). Traditionally, such methods for hard-negative mining rely on a pre-labeling step, where batches are pre-constructed in an offline-scan, and then consumed during training. Unfortunately, such a pre-labeling approach does not scale well as the size of the training data increases. To remedy this, we introduce Ohm, a simple online hard-negative mining procedure that can run in line with traditional streaming data pipelines. An overview of the Ohm approach is given in Algorithm 1.

Ohm consists of several key stages each augmenting a data pipeline. In the first stage, samples are collected into an initial buffer B using a stateful map. When the size of the initial buffer exceeds the update window size, then a parametric clustering method C_ϕ is fit on the samples from B. Note that this process happens per-device, and only with the samples yielded to each device, leading to non-blocking, and non-communicative behavior. Future version of Ohm could , however, communicate the C_ϕ model, leading to all-reduce like behavior with reduced overhead. Once C_ϕ had been trained, C_ϕ is used in a streaming fashion to assign labels to each sample. Finally, reservoir sampling (Vitter, 1985) is used to sample batches of samples from each cluster group as they become available. This leads to batches which are generally closer semantically, even in the presence of a poor clustering algorithm C_ϕ. Since the representations that we are using change, we periodically update C_ϕ (every 10,000 steps) using the buffer B. Ohm thus solves a practical problem: training on GPUs with less VRAM precludes the use of larger (more effective) batch sizes, and leveraging Ohm claws back some of that performance loss.

3.1.4. Reformulation Up-Sampling

In addition to contrastive learning we further over-sample interactions containing reformulations during training of both the transducer and re-scorer. This approach can help to emphasize loss from reformulation samples, without introducing additional overhead. In our experiments, we empirically find an over-sampling ratio of 1:5 (1 reformulation to every 5 standard samples) to be effective (See Table 1).

Algorithm 1 Ohm: Online Hard Negative Mining

Require: stream
Ensure: Reordered stream

Initialize state with zero index and an empty feature array

▷ Step 1: Update clusters with streaming scan (stateful map)
stream ← stream.scan(initial_state, update_clusters)

▷ Step 2: Generate cluster labels for each sample
stream ← stream.map(generate_labels)

▷ Step 3: Group samples by cluster label
stream ← reservoir_sample(dataset)

return batch_processed dataset

procedure update_clusters(buffer, sample)
Add sample to buffer, and trim to update_window_size
if buffer.size() ≥ update_window_size **then**
  Partially fit clusters to buffer
end if
return buffer

procedure generate_labels(sample)
Use clusters to assign sample a cluster label
return {...sample, cluster_label}

procedure reservoir_sample(stream)
reservoir ← empty list for each cluster
while stream.size() > 0 **do**
  sample ← stream.take()
  if any reservoir[cluster].size() > batch_size **then**
    yield from reservoir[cluster]
  end if
end while

Table 1: Word error rate reduction (WERR) on the ALL dataset when using different reformulation up-sampling rates.

Rate	None	1:10	1:5	1:4	1:3	1:2	1:1
WERR	-	0.23	0.46	0.42	0.39	0.35	0.24

3.2. Student-Teacher Distillation

Our overall system comprises both a student and a teacher ASR system. Our student model is a Conformer-based transducer network (Gulati et al., 2020) - a conventional streaming model i.e. it operates on single-utterance and attends to only past frames in the utterance. During run-time, governed by latency constraints, we use the student model to recognize user queries. These interactions (including reformulations and repeats) are captured – details on how to determine reformulations are described in subsection 4.1. Such interactions are then decoded using a context-aware teacher (discussed in the previous section) to get a recognition hypothesis, which acts as a label for semi-supervised training of student model (Parthasarathi & Strom, 2019).

4. Experimental Design

In this section, we discuss the details used when evaluating our approach on both real-world conversational data, and open semi-synthetic data.

4.1. Datasets

All our transducer models are first pre-trained on 200k+ hours of de-identified ASR data (PRETRAIN) using transducer loss without incorporating any contextual information. We then fine-tune and evaluate our approach on one of the two sources of data below: a closed-source real-world dataset from a conversational assistant, and the recently introduced OD3 dataset (Chan et al., 2024).

4.1.1. Closed Source Data

This dataset consists of de-identified dialogues constructed from real-world interactions with a voice assistant. These dialogues are each constructed around a seed utterance, which is human transcribed. Given the seed utterance, the method pulls in all related conversations occurring 90 seconds before and after it. This step is iteratively applied, amassing additional conversational exchanges as they appear. If this approach yields over five utterances, the interval for gathering conversations is cut down to 45 seconds, and the process is repeated. This shortening of the interval persists until the number of utterances falls below five or the interval narrows to a 15-second minimum. These restrictions are enforced to ensure that utterances in a dialog are a semantically coherent interaction around the same request.

Mining dialogues with reformulations To train and evaluate our system, we additionally detect a subset of dialogues consisting of reformulations. To detect reformulations, we use a text similarity-based approach. To be precise, we use cosine similarity and edit distance between the ASR hypothesis of the seed and context utterances (generated during the user's interaction with the assistant). The dialog is said to contain a reformulation if any {seed, context} pair has a similarity greater than the threshold.

We **train** our context encoder teacher models on 10M de-identified dialogues. Additionally, we upsample dialogues containing user reformulations, by 20%, during training of the transducer i.e. one in every five dialogues has reformulation. For **evaluation**, we only select dialogues containing reformulations and ensure that all utterances in the dialog are human-transcribed. By focusing on dialogues where the user reformulated his query, we ensure that the selected dialog has significant ASR errors (as discussed in section subsubsection 3.1.2) and where contextual signals are expected to be meaningfully related to user queries. We create two datasets for evaluation (1) ALL: All transcribed utterances across all validation dialogues (60K utterances) and (2) REF: A subset of ALL containing only utterances that

lead to user reformulations of the query (8.5K utterances).

To **distill** the knowledge of the teacher model to the student model, we use closed-source dialogues containing reformulations. Dialogues used for distillation are distinct from those used for training teacher models and we don't require seed utterances to be human transcribed. Such dialogues are fed to a context-aware teacher model to obtain transcription for seed utterances. This single utterance audio-text pair (approximately 25,000 hours) is then used for training the student model. We refer to this as **s**emi-**s**upervised single utterance **r**eformulation **d**ataset (SSRD).

4.1.2. OD3

Following Chan et al. (2024), we further evaluate our models on the open directed dialogue dataset (OD3). The OD3 dataset is a semi-synthetic dataset, where human-generated task-oriented dialogues from several popular data sets are augmented with LLM-generated conversational errors and computer-generated TTS audio. OD3 contains 620K turns of audio (approximately 1,172 hours).

4.2. Model Details

For our experiments, we use transducer architecture, with Conformer (Gulati et al., 2020) as the audio encoder. We experiment with two different teacher architectures: (i) 200M parameters: 17 conformer blocks and attention size of 1024 (ii) 1B parameters: 18 conformer blocks and attention size of 1536. Each conformer block is composed of four modules - multi-head attention and convolutional modules are sandwiched between two feed-forward modules. The convolutional module has a kernel size of 30 frames. Before conformer blocks, we use a pre-processing block consisting of two convolution layers, which takes in features at a 30ms frame rate, and has a kernel size of 5 and stride of 3.

Our student model has 1B parameters and consists of 18 conformer blocks with an attention size of 1536. However, for student models we restrict the attention module to only attend to past frames - this ensures user-perceived latency is minimal. For the prediction network, we use a two-layer LSTM network with 1024 hidden dimensions and a vocabulary of 4,000 tokens.

4.3. Training details

Our teacher model is pre-trained using the PRETRAIN dataset for 500K iterations, using a per-gpu batch size ranging from 32 to 1, depending on the length of the sequence (sequences are batched to maximally use GPU memory) across either 64 P100 GPUs (for 200M model) or 64 A100 GPUs (for 1B model). We pre-train using an Adam optimizer - we linearly increase the learning rate for 5000 steps and thereafter decrease it proportionally to the inverse square root of the step (as per schedule described in (Vaswani et al., 2017)), and use magnitude-based gradient clipping with a value of 10.

Table 2: WER improvements on the ALL and REF datasets for the teacher model. LE: Learned Embeddings, AE: Audio Embeddings, FC: Feature Concatenation, WERR: Word Error Rate Reduction.

Model	Audio Context	Text Context	CLC + Ohm	WERR (↑) ALL	WERR (↑) REF
Baseline (200M)	-	-	-	-	-
Teacher (200M)	FC	-	-	3.73	7.04
	AE/HuBERT	-	-	1.94	4.04
	AE/Transducer	-	-	2.77	4.97
	AE/Transducer	LE	-	3.32	6.70
	-	BERT	-	0.69	2.66
	-	LE	-	2.63	5.66
	FC	LE	-	4.70	8.08
	FC	LE	✓	**6.91**	**9.58**
Teacher (1B)	-	-	-	0.55	2.89
	FC	LE	-	**5.53**	**9.24**

We then fine-tune our teacher models for 150k steps, using an Adam optimizer with gradient clipping, featuring a learning rate decay schedule that starts at $1e^{-8}$, holds at $1e^{-5}$, and decays to $1e^{-6}$, with the clipping norm set to 0.3, and a schedule policy that adjusts the learning rate at 20K, 80K, and 600K training steps. In addition, we apply dynamic L2 regularization to the Multi-head self-attention layers of the conformer using a PiecewiseConstantDecay scheduler that increases the regularization factor at training steps 15K and 30K (calculated as $2 \times$ number of conformer blocks $\times$ warmup steps), with the regularization values set to $1e^{-6}$, $5e^{-6}$, and $1e^{-5}$ at these intervals.

For models trained with contrastive learning, we use a set of 32 learned clusters for hard-negative mining, with a buffer size of 4096 for the online reservoir sampling. We leverage the BIRCH (Zhang et al., 1997) clustering algorithm, over the embeddings of X^p. In the future, we intend to explore additional clustering algorithms and leverage better distance functions for the Ohm mining approach. For the hyper-parameters of the PF-CLC loss, we follow the parameters in Chan et al. (2024). Our student model is pre-trained using PRETRAIN dataset for 400K iterations with 64 A100 GPUs and a per-gpu batch size ranging from 128 to 1. For model distillation, we use both SSRD and PRETRAIN data, sampled at differing ratios, and standard ASR transducer loss.

5. Results and Discussion

As discussed in section 4, we evaluate our system on both closed-source data and the OD3 dataset. In general, we use both standard word error rate (WER, ↓) and relative word error rate improvement (WERR, ↑) to evaluate our system.

5.1. Teacher Performance

Our overall results for the teacher model on the ALL and REF are given in Table 2. We can see that our system, combining the feature-concatenation audio context (subsection 3.1), learned text context (subsection 3.1), and CLC/Ohm losses (subsubsection 3.1.2), outperforms the baseline model by

Table 3: Results on OD3 (overall and repeat/rephrase inducing) using the 200M model. WER (↓): Word Error Rate, BERT-S (↑): Bert Score. B: Basline, CX: Context, C: CLC Loss, O: Ohm.

Model	Overall		Repeat/Rephrase	
	WER (WERR)	BERT-S	WER (WERR)	BERT-S
B	11.90 (-)	0.9711	19.09 (-)	0.9402
B/CX	11.13 (6.47%)	0.9762	16.17 (15.29%)	0.9690
B/CX/C	8.99 (24.4%)	0.9812	13.81 (27.65%)	0.9737
B/CX/C/O	**8.73 (26.2%)**	**0.9817**	**13.21 (30.8%)**	**0.9771**

Table 4: Zero-shot results on OD3 for several open-source models - Whisper (Radford et al., 2023), Conformer (Gulati et al., 2020), Wav2Vec 2 (Baevski et al., 2020), Streaming Conformer (Tsunoo et al., 2021), CLC (Chan et al., 2024). Models in this table are not directly comparable (trained on differing data, setups, hyperparameters, optimizers etc.), but serve as a benchmark for performance on OD3 under several varying setups. WER (↓): Word Error Rate, BERT-S (↑): Bert Score

Model	Overall		Repeat/Rephrase	
	WER	BERT-S	WER	BERT-S
Whisper S (200M)	11.24	0.9775	14.17	0.9727
Whisper L (1.3B)	8.51	0.9852	12.37	0.9792
Conformer (100M, Librispeech)	19.26	0.9612	22.19	0.9571
Wav2Vec 2 (433M, Librispeech)	19.41	0.9582	22.03	0.9544
Streaming Conformer (45M)	14.38	0.9701	16.70	0.9665
CLC (200M)	8.99	0.9812	13.81	0.9737
Our model (200M)	8.73	0.9817	13.21	0.9771

up to 9.58% on the REF and up to 6.91% on the ALL dataset. These trends hold across model sizes, as our context-enabled model has similar improvement in both the 200M and 1B cases, implying such improvements are model-size independent. Note that for ASR models, 1B parameters is generally considered quite large, given the challenging latency and run-time requirements for ASR applications.

Impact of audio context: Both the approaches of incorporating audio context (feature concatenation and audio embeddings) improve over a baseline system that doesn't use contextual signals. Improvements due to feature concatenation are larger ($> 7\%$ on the reformulation test set), which is not unexpected; by concatenating features we allow the model to attend to all context frames as necessary, as opposed to attending to summary vectors ($N = 8$, in our experiments) coming from multi-headed attentive pooling. Among the two approaches of extracting audio embeddings, we find using a transducer encoder as a context encoder is marginally better, likely as the embeddings are "on-policy" for the trained model, as opposed to coming from an external embedding model.

Impact of text context: In general, we find that encoding text via learned embeddings as opposed to summary vector by BERT encoder is more beneficial. This is aligned with the observation made above for audio context embeddings - again, likely because the latent space is "on-policy" and trained specifically on in-domain data. We also see that textual context under-performs audio context, likely due to incorrect text content from the teacher model.

Table 5: Ablation experiments on the teacher model (200M). WERR: Word Error Rate Reduction.

Context Type				WERR (↑)	
Past	Future	Audio	Text	ALL	REF
-	-	-	-	-	-
✓	-	✓	-	1.52	3.35
✓	-	-	✓	1.24	1.5
✓	-	✓	✓	2.90	5.08
-	✓	✓	✓	3.60	7.39
✓	✓	✓	✓	**4.70**	**8.08**

Table 6: WER improvements on the ALL and REF datasets for the teacher model with CLC/Ohm and **context-aware teacher model as baseline**. Note: Baseline is different from Table 1 to ensure comparable training setup. WERR: Word Error Rate Reduction.

Model	Context	CLC	Ohm	WERR (↑)	
				ALL	REF
Teacher (200M)	✓	-	-	-	-
	✓	✓	-	6.09	8.68
	✓	✓	✓	**6.88**	**10.60**

Combining Context Types: We get the best performance when both audio and text contexts are combined (compared to adding two modalities individually). Interestingly, the 200M model with context is significantly better than the 1B model without contextual signals, highlighting the efficacy of our proposed approach in modeling implicit context signals. On OD3, we can see that adding both context types leads to a 6.47% performance improvement, tracking similarly to the performance improvements seen in the ALL dataset.

Causal vs. Non-Causal Context: In Table 5, we ablate the types of context that we show to the model. We observe that injecting non-causal ("future") context *during training* provides relative WERR of 7.39% as opposed to 5.08% on the REF dataset (as well as improvements on the ALL dataset), indicating that future context is significantly more important when correcting user reformulations. This is likely due to the fact that user reformulation is a "future signal" i.e. it follows the utterance that caused the error.

Implicit Context Learning: We can see that learning from the implicit context in the model is important for understanding and correcting dialog errors. As shown in Table 6, Adding CLC and Ohm to the baseline model leads to significant improvement in the overall performance, particularly on the REF dataset (so much that it enables a 200M parameter model to outperform a 1B parameter model without such losses). On the OD3 dataset (Table 3), the performance is even more distinct, with learning from implicit context leading to up to a 26.6% relative improvement over a baseline non-context model. In addition, zero shot comparison with other open source benchmark models is shown in Table 4.

5.2. Distilling knowledge to student model

Table 7 shows the performance of our model when distilled to a context-free student model. We can see that in all

Table 7: Table showing the impact of distillation from a teacher model trained with implicit/explicit context. WERR: Word Error Rate Reduction. DE: Distillation Efficiency.

Model	SSRD weight	% Params Adapted	WERR (↑) / DE (↑) ALL	REF
Student (1B)	-	-	-	-
+Distillation	100	20	1.38 / 19.97%	**3.06 / 31.94%**
	20	20	**1.76 / 25.47%**	1.2 / 12.52%
	50	100	1.51 / 21.85%	2.95 / 30.79%

Table 8: WERR when normalized by the domain (instead of by-utterance) on the ALL dataset. WERR (↑): Word Error Rate Reduction. SERR (↑): Sentence Error Rate Improvement.

Model	Context	CLC	Ohm	WERR	SERR
Teacher (200M)	✓	-	-	-	-
	✓	✓	-	3.75	2.40
	✓	✓	✓	**6.64**	**7.79**

cases, the student model distilled from a context-trained model achieves superior performance. We also evaluate the distillation efficiency (DE) of the models – how much of the WER gains of the teacher model were retained during distillation. It is interesting to note that when leveraging the SSRD dataset, only 20% of the parameters in the model are necessary during the distillation process to achieve the same WERR, compared to when less reformulation data is used (see subsection 4.3), indicating that the using the pre-trained teacher model with context not only is more accurate, but can be more efficient as well.

5.3. Tail-Distribution Performance

While overall WER is an important measure, many times, a strong indicator of user satisfaction is performance on a wide range of queries on different topics (Such as home automation, calling/messaging and shopping). In Table 8, we present WERR and SERR (Sentence Error Rate Improvement) when the WER is computed on each topic independently, and then averaged instead of being averaged over all utterances (independent of domain, i.e. Table 8 makes the assumption that all domains are equally likely). From this, we can see that while our non-context models perform well on the most common utterances, the contrastive models lead to significant improvements in less-common domains in our ALL dataset, including queries categorized into shopping (82.86% WERR), calling/messaging tasks (73.7% WERR), and music request tasks (36.8% WERR), all of which often need contextual disambiguation. On the other hand, while still in the long tail of the dataset, our approach performs worse than the baseline on home automation tasks (-22.68% WERR), one of the less diverse tasks that requires less contextual disambiguation. In such cases, our model may be relying more on the context, than the target utterance: leading to decreased performance. It remains interesting for future work to explore how we can dynamically trade off between context clues (for challenging utterances), and non-context learning (for utterances that don't require contextual disambiguation).

6. Conclusion

This work introduces a framework that improves ASR in dialog systems through a dual-model approach to contextual learning: a context-aware teacher model that improves learning through explicit and implicit context signals, and a distilled student model that maintains efficiency during inference without context reliance. We achieve significant WER reductions, up to 9.58% on real-world datasets and 26.6% on the OD3 dataset, with the student model maintaining up to 33% of the reduction without context across the distillation process. The enhancements observed, particularly for rare words and diverse user queries, indicate a path toward more robust and satisfying conversational experiences, notably, the pronounced gains for tail queries suggests that our approach can significantly improve performance on less common tasks. Future directions for this work involve exploring the dynamic adjustment of the relative importance of context versus the target utterance based on their predicted utility. This could potentially unlock even greater improvements in ASR performance, paving the way for more intelligent and adaptable conversational AI systems.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work. Automatic speech recognition technology enhances accessibility, education, healthcare, legal processes, customer service, workplace productivity, language preservation, global connectivity, media accessibility, and safety across various societal sectors. While such impact is largely positive, it is important to recognize the impact of self-learning systems for automatic speech recognition on greater discussions in privacy and security, which are well discussed in related work (Chennupati et al., 2022; Aloufi et al., 2021).

References

Aloufi, R., Haddadi, H., and Boyle, D. Configurable privacy-preserving automatic speech recognition. In Hermansky, H., Cernocký, H., Burget, L., Lamel, L., Scharenborg, O., and Motlícek, P. (eds.), *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pp. 861–865. ISCA, 2021. doi: 10.21437/Interspeech.2021-1783. URL https://doi.org/10.21437/Interspeech.2021-1783.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural*

Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

Biadsy, F., Chen, Y., Zhang, X., Rybakov, O., Rosenberg, A., and Moreno, P. J. A scalable model specialization framework for training and inference using submodels and its application to speech model personalization. *ArXiv preprint*, abs/2203.12559, 2022. URL https://arxiv.org/abs/2203.12559.

Buciluǎ, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541, 2006.

Chan, D. M. and Ghosh, S. Content-context factorized representations for automated speech recognition. In *Interspeech*, 2022.

Chan, D. M., Ghosh, S., Chakrabarty, D., and Hoffmeister, B. Multi-modal pre-training for automated speech recognition. In *ICASSP*, 2022.

Chan, D. M., Ghosh, S., Rastrow, A., and Hoffmeister, B. Domain adaptation with external off-policy acoustic catalogs for scalable contextual end-to-end automated speech recognition. In *ICASSP 2023-2023*, pp. 1–5. IEEE, 2023.

Chan, D. M., Ghosh, S., Tulsiani, H., Rastrow, A., and Hoffmeister, B. Task oriented dialogue as a catalyst for self-supervised automatic speech recognition. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11806–11810, 2024. doi: 10.1109/ICASSP48485.2024.10447164.

Chang, F.-J., Liu, J., Radfar, M., Mouchtaris, A., Omologo, M., Rastrow, A., and Kunzmann, S. Context-aware transformer transducer for speech recognition. In *2021 ASRU*, pp. 503–510. IEEE, 2021.

Chang, K.-W., Tseng, W.-C., Li, S.-W., and Lee, H.-y. Speechprompt: An exploration of prompt tuning on generative spoken language model for speech processing tasks. *ArXiv preprint*, abs/2203.16773, 2022. URL https://arxiv.org/abs/2203.16773.

Chang, S.-Y. et al. Context-aware end-to-end asr using self-attentive embedding and tensor fusion. In *ICASSP 2023-2023*, pp. 1–5. IEEE, 2023.

Chen, Z., Jain, M., Wang, Y., Seltzer, M. L., and Fuegen, C. Joint grapheme and phoneme embeddings for contextual end-to-end ASR. In Kubin, G. and Kacic, Z. (eds.), *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pp. 3490–3494. ISCA, 2019. doi: 10.21437/Interspeech.2019-1434. URL https://doi.org/10.21437/Interspeech.2019-1434.

Chennupati, G., Rao, M., Chadha, G., Eakin, A., Raju, A., Tiwari, G., Sahu, A. K., Rastrow, A., Droppo, J., Oberlin, A., Nandanoor, B., Venkataramanan, P., Wu, Z., and Sitpure, P. Ilasr: privacy-preserving incremental learning for automatic speech recognition at production scale. In *KDD 2022*, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 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)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Dingliwal, S., Sunkara, M., Ronanki, S., Farris, J., Kirchhoff, K., and Bodapati, S. Personalization of ctc speech recognition models. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 302–309. IEEE, 2023.

Duarte-Torres, S., Sen, A., Rana, A., Drude, L., Gomez-Alanis, A., Schwarz, A., Rädel, L., and Leutnant, V. Promptformer: Prompted conformer transducer for asr. *ArXiv preprint*, abs/2401.07360, 2024. URL https://arxiv.org/abs/2401.07360.

Futami, H., Inaguma, H., Mimura, M., Sakai, S., and Kawahara, T. Distilling the knowledge of bert for ctc-based asr. *ArXiv preprint*, abs/2209.02030, 2022. URL https://arxiv.org/abs/2209.02030.

Gao, H., Ni, J., Qian, K., Zhang, Y., Chang, S., and Hasegawa-Johnson, M. Wavprompt: Towards few-shot spoken language understanding with frozen language models. *ArXiv preprint*, abs/2203.15863, 2022. URL https://arxiv.org/abs/2203.15863.

Gourav, A., Liu, L., Gandhe, A., Gu, Y., Lan, G., Huang, X., Kalmane, S., Tiwari, G., Filimonov, D., Rastrow, A., et al. Personalization strategies for end-to-end speech recognition systems. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7348–7352. IEEE, 2021.

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

Gulati, A., Qin, J., Chiu, C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., and Pang, R. Conformer: Convolution-augmented transformer for speech recognition. In Meng, H., Xu, B., and Zheng, T. F. (eds.), *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event,*

Shanghai, China, 25-29 October 2020, pp. 5036–5040. ISCA, 2020. doi: 10.21437/Interspeech.2020-3015. URL https://doi.org/10.21437/Interspeech.2020-3015.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *stat*, 1050:9, 2015.

Hori, T., Moritz, N., Hori, C., and Roux, J. L. Transformer-based long-context end-to-end speech recognition. In Meng, H., Xu, B., and Zheng, T. F. (eds.), *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pp. 5011–5015. ISCA, 2020. doi: 10.21437/Interspeech.2020-2928. URL https://doi.org/10.21437/Interspeech.2020-2928.

Hori, T., Moritz, N., Hori, C., and Roux, J. L. Advanced long-context end-to-end speech recognition using context-expanded transformers. In Hermansky, H., Cernocký, H., Burget, L., Lamel, L., Scharenborg, O., and Motlícek, P. (eds.), *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pp. 2097–2101. ISCA, 2021. doi: 10.21437/Interspeech.2021-1643. URL https://doi.org/10.21437/Interspeech.2021-1643.

Hsu, W.-N., Tsai, Y.-H. H., Bolte, B., Salakhutdinov, R., and Mohamed, A. Hubert: How much can a bad teacher benefit asr pre-training? In *ICASSP*, 2021.

Huang, M., You, Y., Chen, Z., Qian, Y., and Yu, K. Knowledge distillation for sequence model. In Yegnanarayana, B. (ed.), *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, pp. 3703–3707. ISCA, 2018. doi: 10.21437/Interspeech.2018-1589. URL https://doi.org/10.21437/Interspeech.2018-1589.

Hung, Y.-N., Yang, C.-H. H., Chen, P.-Y., and Lerch, A. Low-resource music genre classification with cross-modal neural model reprogramming. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Jaech, A. and Ostendorf, M. Personalized language model for query auto-completion. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 700–705, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2111. URL https://aclanthology.org/P18-2111.

Jayanthi, S. M., Kulshreshtha, D., Dingliwal, S., Ronanki, S., and Bodapati, S. Retrieve and copy: Scaling asr personalization to large catalogs. In *EMNLP 2023*, 2023.

Kim, H., Na, H., Lee, H., Lee, J., Kang, T. G., Lee, M., and Choi, Y. S. Knowledge distillation using output errors for self-attention end-to-end models. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pp. 6181–6185. IEEE, 2019a. doi: 10.1109/ICASSP.2019.8682775. URL https://doi.org/10.1109/ICASSP.2019.8682775.

Kim, S. and Metze, F. Dialog-context aware end-to-end speech recognition. In *SLT*, 2018.

Kim, S., Dalmia, S., and Metze, F. Gated embeddings in end-to-end speech recognition for conversational-context fusion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1131–1141, Florence, Italy, 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1107. URL https://aclanthology.org/P19-1107.

Ku, P.-J., Chen, I.-F., Yang, H., Raju, A., Dheram, P., Ghahremani, P., King, B., Liu, J., Ren, R., and Nidadavolu, P. Hot-fixing wake word recognition for end-to-end asr via neural model reprogramming. In *ICASSP 2024*, 2024.

Kudo, T. and Richardson, J. 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*, pp. 66–71, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012.

Kurata, G. and Saon, G. Knowledge distillation from offline to streaming RNN transducer for end-to-end speech recognition. In Meng, H., Xu, B., and Zheng, T. F. (eds.), *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pp. 2117–2121. ISCA, 2020. doi: 10.21437/Interspeech.2020-2442. URL https://doi.org/10.21437/Interspeech.2020-2442.

Li, B., Pang, R., Zhang, Y., Sainath, T. N., Strohman, T., Haghani, P., Zhu, Y., Farris, B., Gaur, N., and Prasad, M. Massively multilingual asr: A lifelong learning solution. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6397–6401. IEEE, 2022.

Lin, R., Liu, S., Yang, M., Li, M., Zhou, M., and Li, S. Hierarchical recurrent neural network for document modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 899–907, Lisbon, Portugal, 2015. Association for

Computational Linguistics. doi: 10.18653/v1/D15-1106. URL https://aclanthology.org/D15-1106.

Liu, B. and Lane, I. Dialog context language modeling with recurrent neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, pp. 5715–5719. IEEE, 2017. doi: 10.1109/ICASSP.2017.7953251. URL https://doi.org/10.1109/ICASSP.2017.7953251.

Masumura, R., Makishima, N., Ihori, M., Takashima, A., Tanaka, T., and Orihashi, S. Hierarchical transformer-based large-context end-to-end asr with large-context knowledge distillation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5879–5883. IEEE, 2021.

Mitra, S. et al. Unified modeling of multi-domain multi-device ASR systems. In *TSD*, 2023.

Mun'im, R. M., Inoue, N., and Shinoda, K. Sequence-level knowledge distillation for model compression of attention-based sequence-to-sequence speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pp. 6151–6155. IEEE, 2019. doi: 10.1109/ICASSP.2019.8683171. URL https://doi.org/10.1109/ICASSP.2019.8683171.

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

Novotney, S., Mukherjee, S., Ahmed, Z., and Stolcke, A. CUE vectors: Modular training of language models conditioned on diverse contextual signals. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3368–3379, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.265. URL https://aclanthology.org/2022.findings-acl.265.

Parthasarathi, S. H. K. and Strom, N. Lessons from building acoustic models with a million hours of speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pp. 6670–6674. IEEE, 2019. doi: 10.1109/ICASSP.2019.8683690. URL https://doi.org/10.1109/ICASSP.2019.8683690.

Pham, H. et al. Combined scaling for zero-shot transfer learning. *Neurocomputing*, pp. 126658, 2023.

Pham, M., Cho, M., Joshi, A., and Hegde, C. Revisiting self-distillation. *ArXiv preprint*, abs/2206.08491, 2022. URL https://arxiv.org/abs/2206.08491.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. Robust speech recognition via large-scale weak supervision. *ArXiv preprint*, abs/2212.04356, 2022. URL https://arxiv.org/abs/2212.04356.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. Robust speech recognition via large-scale weak supervision. In *ICML*, 2023.

Radosavovic, I., Dollár, P., Girshick, R. B., Gkioxari, G., and He, K. Data distillation: Towards omni-supervised learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 4119–4128. IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00433. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Radosavovic_Data_Distillation_Towards_CVPR_2018_paper.html.

Robinson, J. D., Chuang, C., Sra, S., and Jegelka, S. Contrastive learning with hard negative samples. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=CR1XOQ0UTh-.

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

Schwarz, A., He, D., Van Segbroeck, M., Hethnawi, M., and Rastrow, A. Personalized predictive asr for latency reduction in voice assistants. *ArXiv preprint*, abs/2305.13794, 2023. URL https://arxiv.org/abs/2305.13794.

Shenoy, A., Bodapati, S., and Kirchhoff, K. Contextual biasing of language models for speech recognition in goal-oriented conversational agents. *ArXiv preprint*, abs/2103.10325, 2021. URL https://arxiv.org/abs/2103.10325.

Shor, J., Emanuel, D., Lang, O., Tuval, O., Brenner, M. P., Cattiau, J., Vieira, F., McNally, M., Charbonneau, T., Nollstadt, M., Hassidim, A., and Matias, Y. Personalizing ASR for dysarthric and accented speech with limited data. In Kubin, G. and Kacic, Z. (eds.), *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pp. 784–788. ISCA, 2019. doi:

10.21437/Interspeech.2019-1427. URL https://doi.org/10.21437/Interspeech.2019-1427.

Sun, C., Ahmed, Z., Ma, Y., Liu, Z., Pang, Y., and Kalinli, O. Contextual biasing of named-entities with large language models. *ArXiv preprint*, abs/2309.00723, 2023. URL https://arxiv.org/abs/2309.00723.

Tang, J., Kim, K., Shon, S., Wu, F., Sridhar, P., and Watanabe, S. Improving asr contextual biasing with guided attention. *ArXiv preprint*, abs/2401.08835, 2024. URL https://arxiv.org/abs/2401.08835.

Tsunoo, E., Kashiwagi, Y., and Watanabe, S. Streaming transformer asr with blockwise synchronous beam search. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pp. 22–29. IEEE, 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

Vitter, J. S. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1): 37–57, 1985.

Wei, K. et al. Attentive contextual carryover for multi-turn end-to-end spoken language understanding. In *2021 ASRU*, pp. 837–844. IEEE, 2021.

Williams, I., Kannan, A., Aleksic, P. S., Rybach, D., and Sainath, T. N. Contextual speech recognition in end-to-end neural network systems using beam search. In Yegnanarayana, B. (ed.), *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, pp. 2227–2231. ISCA, 2018. doi: 10.21437/Interspeech.2018-2416. URL https://doi.org/10.21437/Interspeech.2018-2416.

Yang, C.-H. H., Li, B., Zhang, Y., Chen, N., Prabhavalkar, R., Sainath, T. N., and Strohman, T. From english to more languages: Parameter-efficient model reprogramming for cross-lingual speech recognition. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Yu, J., Han, W., Gulati, A., Chiu, C., Li, B., Sainath, T. N., Wu, Y., and Pang, R. Dual-mode ASR: unify and improve streaming ASR with full-context modeling. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=Pz_dcqfcKW8.

Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., and Ma, K. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 3712–3721. IEEE, 2019. doi: 10.1109/ICCV.2019.00381. URL https://doi.org/10.1109/ICCV.2019.00381.

Zhang, T., Ramakrishnan, R., and Livny, M. Birch: A new data clustering algorithm and its applications. *Data mining and knowledge discovery*, 1:141–182, 1997.

Zhao, D., Sainath, T. N., Rybach, D., Rondon, P., Bhatia, D., Li, B., and Pang, R. Shallow-fusion end-to-end contextual biasing. In Kubin, G. and Kacic, Z. (eds.), *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pp. 1418–1422. ISCA, 2019. doi: 10.21437/Interspeech.2019-1209. URL https://doi.org/10.21437/Interspeech.2019-1209.

A. Insertions/Deletions/Substitutions in CLC/Ohm

We can further break down the performance in Table 6 in terms of insertions, deletions and substitutions, which is given in Table A.1. We can see that adding CLC loss significantly improves the rate of deletion compared to baseline models. Unfortunately, this comes at the cost of improvement in substitution and insertion. CLC, instead of doing the best job of disambiguating generated tokens, focuses on recall as opposed to precision. Ohm improves the disambiguation, as at the cost of deletions: more tokens are dropped, but the tokens that are preserved are more accurate.

Table A.1: WERR when normalized by the domain (instead of by-utterance) on the ALL dataset. WERR (↑): Word Error Rate Reduction. SERR (↑): Sentence Error Rate Improvement. INSR: Relative Insertion rate. SUBR: Relative Substitution Rate. DELR: Relative Deletion Rate

Model	Context	CLC	Ohm	WERR	SERR	SUBR	INSR	DELR
Teacher (200M)	✓	-	-	-	-	-	-	-
	✓	✓	-	3.75	2.40	1.56	0.82	13.65
	✓	✓	✓	**6.64**	**7.79**	**3.19**	**1.70**	**6.18**