Interspeech 2024
1-5 September 2024, Kos, Greece

Speculative Speech Recognition by Audio-Prefixed Low-Rank Adaptation of Language Models

Bolaji Yusuf[1,2,3], *Murali Karthick Baskar*[1], *Andrew Rosenberg*[1], *Bhuvana Ramabhadran*[1]

[1]Google Inc., USA
[2]Bogazici University, Turkey
[3]Brno University of Technology, Speech@FIT, Czechia

{iyusuf}@fit.vut.cz, {mkbaskar,rosenberg,bhuv}@google.com

Abstract

This paper explores speculative speech recognition (SSR), where we empower conventional automatic speech recognition (ASR) with speculation capabilities, allowing the recognizer to run ahead of audio. We introduce a metric for measuring SSR performance and we propose a model which does SSR by combining a RNN-Transducer-based ASR system with an audio-prefixed language model (LM). The ASR system transcribes ongoing audio and feeds the resulting transcripts, along with an audio-dependent prefix, to the LM, which speculates likely completions for the transcriptions. We experiment with a variety of ASR datasets on which show the efficacy our method and the feasibility of SSR as a method of reducing ASR latency.

Index Terms: low-latency speech recognition, speculative speech recognition, prefix language model, low-rank adaptation
Index Terms: Some keywords

1. Introduction

The experience of users interacting with an automatic speech recognition (ASR) system is colored by its latency—how quickly it is able to respond to user requests—in addition to its accuracy. A system which can respond quickly to user queries is generally preferred to a slower one with similar accuracy.

There has therefore being considerable effort towards improving ASR latency, such as using lightweight, fully-causal or limited-context encoders [1–4], and using modified training objectives such as timing penalties [5] and FastEmit [6] which encourage early output symbol emission to counteract the tendency of limited-context models to delay emission until enough context has been accumulated to make a confident decision. These methods aim to make the latency as close to zero as possible without incurring significant degradation recognition accuracy, i.e., the best case for these approaches is that the model finishes transcription just as the user finishes speaking. However, ASR is only the first step in user interaction and it is often followed by some form of natural language processing (NLP) such as information retrieval or machine translation or spoken language understanding. Therefore, even if ASR latency were to reach zero, the overall end-to-end latency experienced by the user would still above zero.

Prefetching [7, 8] provides a template for reducing the end-to-end latency further. The method hinges on the observation that there is a delay between the an ASR system's emission of the last output symbol and being able to confidently determine that the utterance has ended. ASR hypotheses are sent downstream as soon as a token is emitted without awaiting the end-of-utterance confirmation, and the downstream computation commences immediately. Thus, the endpointing latency is mitigated in exchange for extra computational overhead.

In this paper, we tackle speculative speech recognition (SSR)[1], the problem of accurately generating the *full* transcription *before* the user has finished speaking. Being able to solve this problem would allow any downstream NLP operations to be initiated earlier, and therefore further reduce end-to-end latency. Conceptually, this problem has two parts: transcription and speculation, where the former corresponds to the generation of textual tokens whose corresponding speech has actually been uttered and the latter corresponds to the generation of tokens which have no corresponding input speech.

Schwarz et. al. [11] recently proposed a method for SSR. Their approach—which we take as baseline in this work—uses a hybrid of an RNN-Transducer (RNN-T) [12] ASR system and a pretrained language model (LM), where the RNN-T transcribes the incomplete spoken utterance and the resulting hypotheses are fed into the LM to speculate likely completions.

The ASR-LM hybrid is however limited in that the LM is pretrained for generic text completion, and, consequently, does not account for the idiosyncrasies of operating on ASR output. Specifically, it doesn't account for the fact that its input is a *hypothesis* from an ASR system and may thus contain errors; it also does not consider information contained in the audio signal which may be lost in the transcript, but would nevertheless be useful for speculation. Therefore, we propose a modified scheme which prepends an audio-dependent soft prompt [13] to the ASR hypothesis as input to the LM. We further finetune the LM with low-rank (LoRA) adapters [14] which are trained—along with the soft prompt—to speculate the ideal suffix tokens to complete the ASR hypothesis, thereby allowing the model to have knowledge of both its acoustic context and the possibility of errors from ASR when making decisions.

As the main contributions of this work, we propose:

- A Conformer-Transformer hybrid model which uses an audio-conditioned prefix LM for SSR, with experiments showing the efficacy of the proposed system on public datasets, and its superiority to purely LM-based speculation.
- An edit-distance-based alignment procedure to get training labels for error-aware speculative ASR systems.
- Suffix Oracle Word Error Rate (SOWER), a metric for measuring the performance of a speculative ASR system, which accounts for various peculiarities of the problem.

2. Methods

In conventional ASR, the goal is to predict a sequence of tokens $Y = (y_1, y_2, \ldots, y_U)$ given a sequence of acoustic inputs

[1]Note that this bears no direct relation to speculative decoding in language modelling [9, 10], which involves using a small language model to speed up inference in a larger model.

10.21437/Interspeech.2024-298

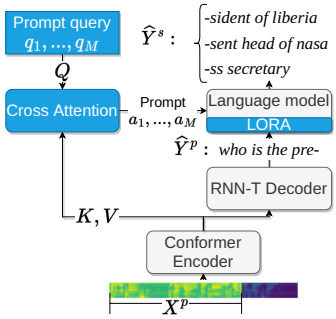

Figure 1: *Illustration of the proposed model with trainable parameters in blue. A Conformer-Transducer ASR model decodes the speech into text. Then a language model is prompted with a prefix computed from the Conformer encoder output to predict likely completions for the partial ASR hypothesis.*

$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$. This generally involves modelling the conditional probability:

$$P(Y|\mathbf{X}) = \prod_{u=1}^{U} P(y_u|\mathbf{X}, y_{<u}). \quad (1)$$

In speculative ASR, instead of the full acoustic input ($\mathbf{X}$), the input is a prefix comprising its first j frames, $\mathbf{X}^p = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_j) : j < T$, and the goal is to model the *full* output sequence, Y, as if the input were the entirety of $\mathbf{X}$:

$$P(Y|\mathbf{X}^p) = \prod_{u=1}^{U} P(y_u|\mathbf{X}^p, y_{<u}). \quad (2)$$

2.1. Baseline speculative ASR with ASR-LM hybrid

Our approach to solving speculative ASR is based on the hybrid approach from [11], with the LSTMs in the RNN-T and LM replaced by a pretrained Conformer-Transducer [15] ASR model and a pretrained Transformer language model [16]. The Transducer is used to transcribe the spoken prefix $\mathbf{X}^p$ into $Y^p = (y_1, \ldots, y_r)$, and the LM is used to predict the suffix $Y^s = (y_{r+1} \ldots, y_U)$ by conditioning on the transcription. This essentially parameterizes (2) thus:

$$P(Y|\mathbf{X}^p) = \underbrace{\prod_{u=1}^{r} P_\phi(y_u|\mathbf{X}^p, y_{<u})}_{\text{Transcription by RNN-T}} \cdot \underbrace{\prod_{u=r+1}^{U} P_\theta(y_u|y_{<u})}_{\text{Speculation by LM}}, \quad (3)$$

where ϕ denotes the parameters of the Transducer model and θ denotes the parameters of the language model.

2.2. Audio-aware speculative ASR

Simply stacking the Transducer and the LM as in Section 2.1 implicitly assumes that the transcription is an efficient enough summary of the input speech for the purposes of speculating the suffix, i.e., that $P(Y^s|X^p, Y^p) = P(Y^s|Y^p)$. However, the speech signal itself contains information such as speaker, channel or domain clues which could be useful for better constraining the language model output. Furthermore, since the Transducer objective considers all possible alignments (in contrast to hybrid models which use forced alignment) , there is only a loose correspondence between the input frame and the output token (j and r respectively in Section 2.1), meaning that the ASR can choose to transcribe sounds later than they are uttered. We argue, therefore, that it would be useful to condition the suffix generation on a representation of the audio signal.

To this end, we add a fixed-length representation, $\mathbf{A}^p = (\mathbf{a}_1, \ldots, \mathbf{a}_M)$, of the audio as a prefix to the language model. $\mathbf{A}^p$ is the output of a multihead attention layer whose keys and values are projections of the Transducer's Conformer encoder output, and whose queries are projections a sequence of M trainable vectors, $\mathbf{Q} = (\mathbf{q}_1, \ldots, \mathbf{q}_M)$. Thus, when the Transformer speculates the uth overall output token, its input is:

$$c_u = \big(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_M, \mathbf{e}[y_1], \mathbf{e}[y_2], \ldots \mathbf{e}[y_{u-1}]\big), \quad (4)$$

where $\mathbf{e}[\cdot]$ denotes the LM's input embedding lookup. Although it is possible in theory to use other speech representations, we choose the Conformer output since we get it for free as part of the ASR computation. Moreover, we know that it is rich in lexical content since it is an encoding that is used directly for ASR.We use a fixed-length summary of the encoding as prefix because doing so reduces (when $j > M$), and keeps fixed, the computational cost of the subsequent LM decoding compared to using the encoding directly. Ultimately, instead of (3), we have the following parameterization:

$$P(Y|\mathbf{X}^p) = P_\phi(Y^p|\mathbf{X}^p) \prod_{u=r+1}^{U} P_{\phi_e,\theta,Q,\zeta}(y_u|y_{<u}, \mathbf{X}^p), \quad (5)$$

where ζ denotes the multihead attention parameters. Our training process will involve keeping the first term on the right hand side fixed, and learning parameters which maximize the second term—$P_{\phi_e,\theta,Q,\zeta}(Y^s|Y^p, \mathbf{X}^p) \coloneqq \prod_{u=r+1}^{U} P_{\phi_e,\theta,Q,\zeta}(y_u|y_{<u}, \mathbf{X}^p)$.

2.3. Alignment and finetuning for speculation

With the parameters of the Conformer-Transducer fixed, we add LoRA layers to each Transformer layer in the LM. The LoRA parameters are trained along with the LM's tied embedding-softmax layer, the cross-attention parameters and query vectors to maximize the log-likelihood of the correct suffix.

We do the finetuning on a dataset of audio-text transcription $(\mathbf{X}, Y)$ pairs. For each training sample, we first get $\mathbf{X}^p$ by truncating the last *1 second* of audio. Then we pass the truncated audio to the Transducer-based ASR system to get a hypothesized transcript $\hat{Y}^p$. Next we feed this hypothesis into the language model—along with the prefix from the cross-attention on the encoder output—to predict $Y^s = (y_{r+1}, \ldots, y_U)$—the portion of Y corresponding to the truncated 1 second of audio.

Determining r for training, however, is not as trivial as it may seem. Consider, for example, an utterance whose correct transcription is *"i'd like to call my father"*, but for which the ASR generates the prefix *"i'd line to call ma-"*. Because of the ASR errors, the correct suffix is not clearly defined. It is therefore necessary to first determine what part of the transcription has been covered—possibly erroneously—by the ASR and what part remains to be speculated.

AWSED: We propose solving this problem by Alignment With Subsequence Edit Distance (AWSED). This procedure, illustrated in Figure 2, involves computing the Levenshtein distance (LVD) [17] between $\hat{Y}^p$ and all left-substrings of Y. More concretely, taking the left- and right-substring of Y at some index v respectively as $Y_{:v} \coloneqq (y_1, \ldots, y_v)$ and $Y_{v:} \coloneqq (y_{v+1}, \ldots, y_U)$, then the desired target suffix, $Y^s = Y_{v^*:}$, where:

$$v^* = \arg\min_v \; L(\hat{Y}^p, Y_{:v}), \quad (6)$$

where $L(s_1, s_2)$ is the LVD between the strings s_1 and s_2. Note that rather than having to compute $L(\hat{Y}^p, Y_{:v})$ separately for each v, we need only one run of the dynamic programming algorithm for computing the LVD to get v^* because the last row of the matrix of accumulated costs already contains $L(\hat{Y}^p, Y_{:v})$

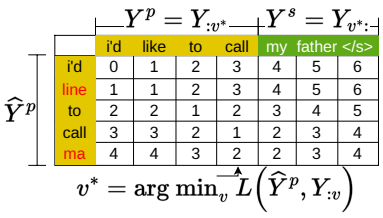

Figure 2: *AWSED procedure for computing the optimal alignment between a hypothesis prefix and a full reference.*

for every v, and we only need to take the $\arg\min$ on this row. In case multiple indices are tied for the $\arg\min$, we pick the leftmost one. In our running example, the AWSED procedure yields the resulting Y^s as *"-my father"*. Had we taken the rightmost $\arg\min$, then Y^s would have been *"-father"*. This choice however does not affect the overall word edit distance from the correct transcription, which is 2 in both cases (1 substitution and 1 insertion for *"i'd line to call ma my father"*, and 2 substitutions for *"i'd line to call ma father"*).

We use stochastic gradient descent with Adam [18] to finetune. For each mini-batch $\mathcal{B}$, we minimize the cross-entropy:

$$J_{\mathcal{B}} = -\sum_{(\mathbf{X},Y)\in\mathcal{B}} \sum_{u=v^*+1}^{U} \log P_{\phi_e,\theta+\Delta\theta,\mathbf{Q},\zeta}(y_u|\hat{Y}^p, \mathbf{X}^p), \quad (7)$$

with respect to the cross-attention parameters (ζ), soft prompt query vectors ($\mathbf{Q}$) and LoRA parameters ($\Delta\theta$).

3. Experiments

3.1. Metrics

SSR differs from conventional ASR in one crucial respect: it has no single correct answer; for a given prefix audio, it is impossible to determine the suffix with perfect certitude. This makes the word error rate (WER) metric unsuitable for out purposes. Moreover WER—or any other metric that is computed over the entire sentence—also encompasses a measure of the prefix transcription accuracy and thus can only be, at best, a diffuse proxy for speculation performance. Hence, we seek a metric which isolates the accuracy of the speculated suffix while also accounting for the uncertainty inherent in the task.

One solution would be to treat the task as a language modeling one, and to report the perplexity of the correct suffix. Perplexity however has the drawback that it is not an operational measure, i.e., we can use it to compare systems and say lower is better, but knowing the exact perplexity score says little about the conditions under which the system is usable. Furthermore, perplexity has to be computed with "teacher-forcing", which does not reflect the practical usage of an ASR system.

Instead, we treat speculation as a quasi-retrieval problem and propose an analog of recall-at-k. The metric, which we term suffix oracle word error rate (SOWER) is computed by truncating t seconds from the end of the audio, letting the system hypothesize k suffixes $\hat{Y}^s(1), \ldots, \hat{Y}^s(k)$, and returning the minimal WER between the hypotheses and the correct suffix Y^s:

$$S(t,k) = \min_{i\in[1,k]} \text{WER}(\hat{Y}^s(i), Y^s). \quad (8)$$

This measures how well we expect the model to do if we return the top-k hypotheses . By default, we set $t = 1$ and $k = 8$ in our experiments. Note that, to get the correct suffix for evaluation, we once again use the AWSED procedure described in Section 2.3 on the prefix transcription and the full reference.

3.2. Datasets and model architecture

We use Speechstew [19], an amalgamation of multiple public corpora totalling about 5000 hours, to pretrain the Conformer for transcription. We conduct two sets of experiments, varying the pretraining, finetuinnig and testing data:

Librispeech-only: Here, we use Librispeech LM data [20] for pretraining the language model, finetune on the 960h Librispeech training set, and test on the Librispeech test splits.

Multi-domain: Here, we pretrain the LM with data composed of the Librispeech LM data along with text from the Switchboard [21], TED-LIUM [22, 23] and Wall Street Journal [24] corpora. We finetune on Speechstew, and test on the AMI-IHM [25], Librispeech, Switchboard and TED-LIUM test sets.

The Transducer model has a 100 million parameter, 17-layer Conformer-L [15] encoder with 512-dimensional layers operating on 80-dimensional log-mel filterbank inputs, an LSTM prediction network with one 512-dimensional layer, and a 2-layer feedforward joint network with 512-dimensional intermediate layer and 1024-dimensional output softmax layer corresponding to 1024 Librispeech word-piece [26] targets.

The LM is a 100 million parameter transformer decoder [16] with eight 1024-dimensional layers, each split into 16 attention heads, and tied embedding-softmax with the same 1024 word piece units as the Transducer.

We use $M = 64$ queries (equivalent in length to 2.56 seconds of audio) of 1024 dimensions and a single 1024-dimensional cross-attention layer with four heads for computing the soft-prompt. We set the rank of the LoRA adapters to 10 for the Librispeech experiments and 50 for the multi-domain experiments, resulting in 12 million and 19 million trainable parameters respectively to be finetuned for speculation.

3.3. Speculation systems

We report results for four model configurations:

Pretrained model (PM): This is the our replication of the model from [11] (described in §2.1), which uses the pretrained LM to speculate suffixes without any finetuning or audio prefix. We note that [11] also incorporates a confidence model on top of speculation. Here, we focus only on the speculator itself.

Hypothesis-only finetuned (HO): Similar to PM, this configuration only uses the Transducer hypotheses—without audio encoding—as the LM input. The LM, however, is finetuned for speculation. Instead of the audio-aware prefix, $\mathbf{A}^p$, we use the trainable vectors, $\mathbf{q}_1, \ldots, \mathbf{q}_M$, directly as the (audio-agnostic) prefix and finetune them along with the LM softmax and LoRA layers. This configuration allows us to separate what improvements, if any, come from ASR-error-aware finetuning and what improvements come from using an audio prefix.

Speech-prefix finetuned (SP): This is the full configuration as described in §2.2, which uses an audio-dependent soft prompt.

Speech prefix + Text Injection (ST): This configuration adds text injection [27,28] to SP. While finetuning on paired speech-text data, the trainable parameters are jointly trained on the text used to pretrain the respective LM, so that the LM does not overfit to the paired training data. Since the text-only training mini-batches have no audio input, their cross-attention keys and values and, consequently the LM prompts, are set to 0.

3.4. Librispeech results

First we conduct experiments of the Librispeech test sets, where we truncate the last second of audio from each utterance and speculate them with various systems. This 1 second of audio

Table 1: *SOWER on the Librispeech dev and test sets; "tavg" is the average of the test sets. PM-1w and ST-1w denote SOWER computed on just the first suffix word;* $\Delta_{97.5}$ *and* $\Delta_{2.5}$ *denote respectively the* 97.5 *and* 2.5 *percentile values of the improvement over PM estimated with blockwise bootstrap [29, 30].*

	dev-c	dev-o	test-c	test-o	tavg	$\Delta_{97.5}$	$\Delta_{2.5}$
PM	75.0	79.4	74.4	81.2	77.8	0	0
HO	69.0	73.2	69.0	75.0	72.0	5.6	6.2
SP	64.1	68.7	64.8	69.6	67.2	9.9	11.4
ST	61.0	66.6	61.7	66.9	64.3	12.8	14.2
PM-1w	61.3	67.2	62.0	68.1	65.1	0	0
ST-1w	46.1	51.7	46.4	52.1	49.3	15.3	15.8

Table 2: *Oracle WER on the Librispeech dev and test sets. WERR denotes the percentage tavg WER recovered by speculation, computed as* $100 * \frac{sys-baseline}{topline-baseline}$*, and ST(k=1) refers to using 1-best speculation from ST instead of 8-best.*

	dev-c	dev-o	test-c	test-o	tavg	WERR
Baseline	11.3	16.3	11.6	15.6	13.6	0
PM	8.9	13.9	9.1	13.6	11.4	21.8
HO	8.3	13.1	8.5	12.7	10.6	29.7
SP	7.8	12.5	8.1	12.1	10.1	34.7
ST	7.5	12.3	7.8	11.8	9.8	37.6
ST (k=1)	10.2	15.3	10.4	14.6	12.5	10.9
Topline	2.1	4.7	2.2	4.8	3.5	100

contains an average of around 2 words/utterance (1.98-2.2 depending on the test set) which the systems must speculate.

Table 1 shows SOWER—$S(1, 8)$—on the Librispeech test sets. All the systems, including PM, yield SOWER below 100, i.e., on average, picking the best of top-8 speculated hypotheses is better than not speculating at all. HO significantly outperforms PM, highlighting the positive impact of making the LM ASR-error-aware. SP yields a further 6.7% relative improvement over HO. Finetuning jointly with unpaired text (ST) leads to further improvements, indicating that the SP—and HO—loses some capacity as a language model while fitting the ASR training set (which is much smaller than the unpaired data), and this capacity can be somewhat recovered by joint training. Finally, the table also shows that speculating only the next word is, as expected, easier than predicting the rest of the utterance—with ST-1w in particular averaging SOWER below 50% on single word prediction.

On inspecting the transcriptions and listening to the corresponding audio, we found several cases where the prefix audio ends with the beginnings of a sound, usually a stop consonant, which the ASR does not transcribe. Where the systems without audio prompt speculate semantically-appropriate completions, the ones with audio prompt speculate semantically-appropriate completions which also start with the correct phoneme.

Table 2 shows the Librispeech oracle WER computed over the entire utterances (not just the suffixes), and therefore show the impact of speculation on the whole WER. The speculation methods are compared with two purely Conformer-Transducer systems: a baseline, where each utterance is truncated and no speculation is done, and a topline which sees the entire audio without truncation or need to speculate. Overall, the oracle word error rates of the speculation methods follows the same trends as the SOWER, with the best speculation method (ST) recovering about 37.6% of the gap between the baseline and the topline.

Table 3: *SOWER on various test sets.*

	ami	test-c	test-o	swbd	ted	tavg	$\Delta_{97.5}$	$\Delta_{2.5}$
PM	95.6	80.0	84.2	93.7	89.8	88.7	0	0
HO	88.6	80.2	84.1	89.8	86.3	85.8	1.4	5.8
SP	79.2	69.7	73.7	82.3	76.1	76.2	10.4	15.1
ST	79.6	67.0	71.6	83.2	77.8	75.8	10.2	15.0

Table 2 also shows the oracle WER of the best system, ST, at $k = 1$. Unsurprisingly, the oracle WER (and SOWER although it is not shown) degrades as k is decreased. More interestingly, we see that even at $k = 1$, ST outperforms the baseline. In other words, even one-shot speculation with ST is slightly better than no speculation at all. In fact, we found that all the speculation systems improve upon the baseline with the exception of PM at $k = 1$ (which has SOWER > 100).

3.5. Multi-domain results

In addition to Librispeech, we report results on the AMI, Switchboard and TED-LIUM test sets in order to test generalization of the proposed speculation methods across domains.

Table 3 shows the SOWER across test sets. Here we find that the improvements from HO compared to PM are more subdued. In fact, for the Librispeech test sets, HO is slightly worse than PM. The largest improvements are on AMI, which is not represented in the LM pretraining data, and thus benefits the most from finetuning on the Speechstew training set–which does contain AMI data. SP outperforms HO by a larger margin across all datasets, underscoring the importance of conditioning the LM on audio. Finally, we observe that ST does not outperform SP in terms of SOWER, except on Librispeech. This is likely due to the fact that the Librispeech text is the most represented in the unpaired text used for joint training.

Qualitatively, we found that because the Speechstew data is an amalgamation of various datasets, some of which have transcripts with punctuation and extra tags like "[laughter]", "[noise]" etc., HO learns to frequently speculate these tags even for datasets like Librispeech which are normalized, and thus increases number of errors, while SP does so less frequently. This explains some of the degradation of HO on Librispeech. It also hints at another advantage of having audio conditioning, namely it triggers domain-specific behavior in the LM.

4. Conclusions

In this paper, we've tackled the problem of doing ASR before the getting the input by using an LM to speculate the missing parts. We propose an approach to reducing the entropy in speculation by feeding a fixed-length audio prefix to the LM, and a mechanism for finetuning the LM in the presence ASR-errors. This speculative model is able to correctly retrieve 35.7% of suffixes up to one second ahead of time—and 50.7% when predicting only the next word, showing its viability as a means of achieving negative latency in ASR.

Future work in this direction include finding other ways of reducing the suffix entropy such as incorporating user-specific language models (as done in [11]) or biasing lists, retrieving from related documents [31–33] or simply using better LMs. Furthermore, although we show that we can run ahead-of-audio, fully realizing the latency improvements would require using efficient LMs and LM inference schemes such as [9, 10] especially for larger LMs. Another interesting direction is to incorporate speculation directly into the ASR—the transducer objective, for instance, places no limits on input and output lengths.

5. References

[1] Y. He *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6381–6385.

[2] T. N. Sainath *et al.*, "Two-Pass End-to-End Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2773–2777.

[3] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7829–7833.

[4] G. Strimel, Y. Xie, B. J. King, M. Radfar, A. Rastrow, and A. Mouchtaris, "Lookahead when it matters: Adaptive non-causal transformers for streaming neural transducers," in *International Conference on Machine Learning*. PMLR, 2023, pp. 32 654–32 676.

[5] B. Li *et al.*, "Towards Fast and Accurate Streaming End-To-End ASR," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6069–6073.

[6] J. Yu, Chiu *et al.*, "Fastemit: Low-latency streaming asr with sequence-level emission regularization," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6004–6008.

[7] S.-Y. Chang, B. Li, D. Rybach, Y. He, W. Li, T. N. Sainath, and T. Strohman, "Low latency speech recognition using end-to-end prefetching." in *Interspeech*, 2020, pp. 1962–1966.

[8] B. Li *et al.*, "A Better and Faster End-to-End Model for Streaming ASR," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5634–5638.

[9] Y. Leviathan, M. Kalman, and Y. Matias, "Fast inference from transformers via speculative decoding," in *International Conference on Machine Learning*. PMLR, 2023, pp. 19 274–19 286.

[10] C. Chen, S. Borgeaud, G. Irving, J.-B. Lespiau, L. Sifre, and J. Jumper, "Accelerating large language model decoding with speculative sampling," *arXiv preprint arXiv:2302.01318*, 2023.

[11] A. Schwarz, D. He, M. Van Segbroeck, M. Hethnawi, and A. Rastrow, "Personalized Predictive ASR for Latency Reduction in Voice Assistants," in *Proc. INTERSPEECH 2023*, 2023, pp. 745–749.

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

[13] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3045–3059. [Online]. Available: https://aclanthology.org/2021.emnlp-main.243

[14] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

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

[16] A. Vaswani *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[17] V. I. Levenshtein *et al.*, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[19] W. Chan, D. Park, C. Lee, Y. Zhang, Q. Le, and M. Norouzi, "Speechstew: Simply mix all available speech recognition data to train one large neural network," in *Proc. MLSLP*, 2021.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[21] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: telephone speech corpus for research and development," in *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, ser. ICASSP'92. USA: IEEE Computer Society, 1992, p. 517–520.

[22] A. Rousseau, P. Deléglise, and Y. Esteve, "TED-LIUM: an Automatic Speech Recognition dedicated corpus." in *LREC*, 2012, pp. 125–129.

[23] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Esteve, "TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation," in *Speech and Computer: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 20*. Springer, 2018, pp. 198–208.

[24] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992. [Online]. Available: https://aclanthology.org/H92-1073

[25] W. Kraaij, T. Hain, M. Lincoln, and W. Post, "The AMI meeting corpus," in *Proc. International Conference on Methods and Techniques in Behavioral Research*, 2005.

[26] T. Kudo and J. Richardson, "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*, E. Blanco and W. Lu, Eds. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. [Online]. Available: https://aclanthology.org/D18-2012

[27] P. Wang, T. N. Sainath, and R. J. Weiss, "Multitask Training with Text Data for End-to-End Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 2566–2570.

[28] B. Yusuf, A. Gandhe, and A. Sokolov, "USTED: Improving ASR with a Unified Speech and Text Encoder-Decoder," in *Proceedings of ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE Signal Processing Society, 2022, pp. 8297–8301.

[29] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in asr performance evaluation," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 2004, pp. I–409.

[30] Z. Liu and F. Peng, "Statistical Testing on ASR Performance via Blockwise Bootstrap," in *Proc. Interspeech 2020*, 2020, pp. 596–600.

[31] D. M. Chan, S. Ghosh, A. Rastrow, and B. Hoffmeister, "Domain adaptation with external off-policy acoustic catalogs for scalable contextual end-to-end automated speech recognition," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[32] B. Yusuf, A. Gourav, A. Gandhe, and I. Bulyko, "On-the-Fly Text Retrieval for end-to-end ASR Adaptation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[33] Z. Wu, T. Munkhdalai, P. Rondon, G. Pundak, K. C. Sim, and C. Li, "Dual-Mode NAM: Effective Top-K Context Injection for End-to-End ASR," in *Proc. INTERSPEECH 2023*, 2023, pp. 221–225.