

Transducer-Llama: Integrating LLMs into Streamable Transducer-based Speech Recognition

Keqi Deng^{*†}, Jinxi Guo^{*‡}, Yingyi Ma^{*}, Niko Moritz^{*}, Philip C. Woodland[†], Ozlem Kalinli^{*}, Mike Seltzer^{*}

^{*}Meta AI, USA

{keqi, jinxiguo, yingyima, nmoritz, okalinli, mikeseltzer}@meta.com

[†]Department of Engineering, University of Cambridge, UK

{kd502, pw117}@cam.ac.uk

Abstract—While large language models (LLMs) have been applied to automatic speech recognition (ASR), the task of making the model streamable remains a challenge. This paper proposes a novel model architecture, Transducer-Llama, that integrates LLMs into a Factorized Transducer (FT) model, naturally enabling streaming capabilities. Furthermore, given that the large vocabulary of LLMs can cause data sparsity issue and increased training costs for spoken language systems, this paper introduces an efficient vocabulary adaptation technique to align LLMs with speech system vocabularies. The results show that directly optimizing the FT model with a strong pre-trained LLM-based predictor using the RNN-T loss yields some but limited improvements over a smaller pre-trained LM predictor. Therefore, this paper proposes a weak-to-strong LM swap strategy, using a weak LM predictor during RNN-T loss training and then replacing it with a strong LLM. After LM replacement, the minimum word error rate (MWER) loss is employed to finetune the integration of the LLM predictor with the Transducer-Llama model. Experiments on the LibriSpeech and large-scale multi-lingual LibriSpeech corpora show that the proposed streaming Transducer-Llama approach gave a 17% relative WER reduction (WERR) over a strong FT baseline and a 32% WERR over an RNN-T baseline.

Index Terms—LLMs, online ASR, neural transducer

I. INTRODUCTION

End-to-end (E2E) automatic speech recognition (ASR) simplifies conventional pipeline methods and directly transcribes speech into text [1], [2]. In many real-world scenarios, streaming ASR is needed for low latency. Many E2E approaches [1]–[3] have been developed for such online applications, among which the recurrent neural network transducer (RNN-T) [2] is widely used for streaming operation. However, while the prediction network of the RNN-T has a similar structure to a language model (LM), it does not perform as an explicit LM [3]–[5], which makes it hard to incorporate pre-trained LMs [6]. Several papers [5]–[8] have tried to separate the LM component from the neural transducer, e.g. the non-blank predictor in the Factorized Transducer (FT) [5], [8]. However, a major challenge is that, even with a modularized internal LM component, use of a strong internal LM provides limited ASR improvement on general datasets [7], [9]. Recently, [8] proposed an effective internal LM fusion and training strategy that greatly improves the FT model performance.

Text-based large LMs (LLMs) have achieved great success [10]–[13]. Several recent studies have focused on extending text-based LLMs to handle speech input (speech LLMs), in which speech prompts are prepended to the text sequence and the LLMs are conditioned on the input speech [14], [15]. However, this decoder-only architecture cannot naturally handle streaming as the speech prompts are prepended before text input [16]. In addition, text-based LLMs operate on discrete text units, in which a tokenizer maps the raw text into a token sequence, and LLMs are limited to the vocabulary they were trained on [17]. However, this restricts the flexibility when applying LLMs to speech tasks. For example, the vocabulary size of LLMs can be too large to be used for ASR system training [18]. In addition, the LLM tokenizer may not be optimal for domains for which it wasn't designed [17], [19].

In order to efficiently integrate LLMs into streaming ASR systems, this paper proposes a novel architecture, Transducer-Llama, which is based on the recently proposed FT model [8], using an LLM as a non-blank predictor. To avoid the data sparsity and increased training costs caused by the large LLM vocabulary, this paper introduces an efficient vocabulary adaptation technique, aligning the LLM with a specialized ASR vocabulary. Preliminary experiments show that, compared to a weaker pre-trained LM predictor, using a strong LLM-based non-blank predictor leads to only minor ASR improvements after RNN-T loss training. To address this, we propose a weak-to-strong LM swap strategy, using a weak LM (e.g. stateless predictor) during RNN-T loss training and then replacing it with a strong LLM. After LM replacement, the internal LM-aware minimum word error rate (MWER) loss [8], [20] is applied to further finetune the Transducer-Llama model in order to optimize the integration of the LLM predictor. Therefore, Transducer-Llama provides a framework that naturally applies LLMs to online ASR and enables highly efficient training. Experiments on LibriSpeech [21] and Multi-lingual LibriSpeech (MLS) [22] show the Transducer-Llama can effectively incorporate LLMs to boost ASR accuracy, and also demonstrate superior performance over speech LLMs.

II. RELATED WORK

Given the success of LLMs and the importance of low-latency responses in ASR, recent work has begun to explore online LLM-based speech systems. However, the speech input

[‡]Corresponding author.

Work done while Keqi Deng was an intern at Meta AI.

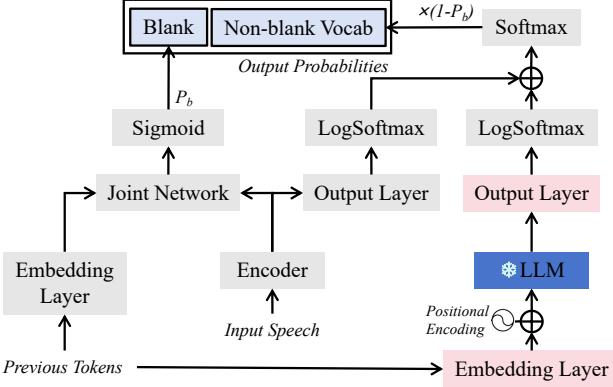


Fig. 1. Illustration of the Transducer-Llama framework. \oplus denotes addition. LLM parameters are fixed (frozen). The output and embedding layers in red are initialised based on the proposed vocabulary adaptation method.

is always prepended before text input as a prompt, making online operation challenging [16]. Hence, online modifications must be made based on known speech-text alignments in order to restrict the speech visible for each text token. [23] uses an external CTC model to provide the alignment, which breaks the flat-start advantage of E2E models. [24] obtains the speech-text alignment on the fly at training via a CTC greedy search, but this reduces training efficiency. [16] instead uses a Wait-k [25] strategy to provide a hard-coded alignment, however, as noted in [16], [26], this strategy is unsuitable for low-latency scenarios. In this paper, the proposed Transducer-Llama provides a natural solution to support LLM-based online ASR, which can be efficiently trained from a flat-start.

Several papers have separated the internal LM from the neural transducer [5]–[8]. This paper uses the modified FT structure based on [8], which can be referred to Fig. 1 (gray parts), except for the LLM part. Separate predictors are used, i.e. a blank predictor and non-blank predictor (internal LM), the non-blank probability P_{nb} is obtained from the sum of the acoustic encoder ($\log P_{ac}$) and internal LM log-probabilities ($\log P_{ilm}$). The outputs of the encoder and blank predictor (e.g. gray embedding layer in Fig. 1) are combined in joint network to produce a single-dimension logit, which is then passed through a sigmoid function to compute blank probability P_b . The non-blank probability P_{nb} is normalized to decouple it from P_b : $P_{nb} = (1 - P_b)(\text{softmax}(\log P_{ac} + \alpha \log P_{ilm})) + \beta \log P_{ilm}$. During decoding, [8] uses a modified score for non-blank tokens:

$$\log((1 - P_b)(\text{softmax}(\log P_{ac} + \alpha \log P_{ilm})) + \beta \log P_{ilm}) \quad (1)$$

where α and β are hyper-parameters.

III. TRANSDUCER-LLAMA

The proposed Transducer-Llama, as shown in Fig. 1, uses the FT architecture and incorporates LLMs as the non-blank predictor at decoding to model causal dependencies, which is the main difference from FT. An embedding layer is used as the blank predictor. The vocabulary adaptation technique, weak-to-strong LM swap strategy followed by MWER training are employed to make Transducer-Llama training efficient and fully utilise the LLM to boost ASR performance.

A. LLM Vocabulary Adaptation

LLMs have large vocabularies not designed for speech systems. This makes training expensive, especially for the RNN-T loss. Therefore, this paper designs a vocabulary adaptation method to adapt the LLM to the ASR system vocabulary. The ASR tokenizer is trained on transcripts from ASR training data, which is typically much smaller than the LLM vocabulary, helping alleviate data sparsity. To achieve efficient training, the LLM can be fixed at training as shown in Fig. 1 while only updating the embedding and output layers.

To align LLMs to the ASR vocabulary, new embedding and output layers need to be employed, since their weight matrices are tied to the vocabulary size, while the Transformer layers of the LLM are kept fixed. To efficiently adapt the vocabulary, while leveraging the original LLM information, this paper initializes these weight matrices (highlighted in red in Fig. 1) based on the original LLM vocabulary, inspired by [27]–[30].

Suppose the weight matrix in the embedding or output layer is $\mathbf{W}^{\text{new}} \in \mathbb{R}^{|V| \cdot d}$, where V is the ASR vocabulary with size $|V|$ and d is the embedding dimension. Denote the LLM vocabulary as U and the corresponding weight matrix as $\mathbf{W}^{\text{ilm}} \in \mathbb{R}^{|U| \cdot d}$. For each token $t_i \in V$, if $t_i \in V \cap U$, then $\mathbf{W}_i^{\text{new}} = \mathbf{W}_j^{\text{ilm}}$, where j is the corresponding index of t_i in the original LLM vocabulary. If $t_i \notin U$, then t_i is tokenized using the LLM tokenizer, and $\mathbf{W}_i^{\text{new}}$ is initialized as the average of the corresponding embeddings. If it cannot be decomposed, it is initialized randomly. Note the newly-trained weight matrices of the embedding and output layer are distinct in this paper.

B. Weak-to-Strong LM Swap

Preliminary experiments showed that when training with the RNN-T loss, introducing a stronger LM as the non-blank predictor results in relatively limited improvements in ASR performance, consistent with previous work [7], [9]. To address this challenge, we propose a weak-to-strong LM swap strategy, using a weak LM during training and replaced by a strong LLM at decoding. This prevents the model from overly relying on the strong LM accurate predictions during training and ensures the acoustic information is fully optimized and utilized. Additionally, this strategy greatly speeds up the training process. In this paper, an embedding layer is used as the weak LM, functioning similarly to a bigram LM and also called a stateless non-blank predictor.

C. Proposed Training and Decoding

We first train a FT model with the weak LM non-blank predictor from scratch. The RNN-T loss is used together with an additional internal LM (ILM) loss [8] to train the internal LM. Once the model has converged, the small internal LM is replaced by an LLM, thereby transforming the FT model into Transducer-Llama. Transducer-Llama can be used for decoding following Eq. 1, in which the LLM predicted probabilities P_{LLM} is considered as P_{ilm} . This weak-to-strong LM swap strategy forces the encoder to fully utilize acoustic information to improve the accuracy of P_{nb} . Compared to integrating LLM directly at RNN-T training stage, it can

fully leverage the capabilities of the LLM to improve ASR performance while also offering fast training speed.

After the LM swap, the internal LM-aware MWER loss (proposed in [8]) is used to train the Transducer-Llama, which directly optimizes the LLM predictor integration with the word-level edit distance and performs sequence discriminative training. The N-best hypotheses are generated according to Eq. 1, where the weights α and β applied during MWER training are also used during inference. It's shown in Section IV that MWER loss can more effectively leverages strong LMs to improve ASR performance, in comparison to RNN-T loss.

With its modular structure, weak-to-strong LM swap following by MWER training, Transducer-Llama shares many similarities with the conventional neural network-hidden Markov model (HMM) hybrid sequence discriminative training strategy [31]–[33] or external LM fusion [34].

IV. EXPERIMENTS

A. Corpus

Experiments were conducted using the LibriSpeech [21] and multi-lingual LibriSpeech (MLS) [22] corpora. Four languages, English (en), French (fr), Italian (it) and Dutch (nl), from the MLS dataset were used as the multi-lingual training data, with respective training audio size of 44.7k hrs, 1.1k hrs, 0.2k hrs, 1.6k hrs. The LibriSpeech LM corpus (800M words) and training data transcripts were used to train the LMs for LibriSpeech experiments. For MLS, the LibriSpeech LM data (En), and the French (146M words), Italian (41M words), and Dutch (46M words) text data from the MLS LM corpus, were used. This multilingual text data, along with training data transcripts, was employed to train the LMs for MLS.

B. Model Descriptions

All ASR systems used a vocabulary of 5000 tokens along with the blank token for both LibriSpeech and MLS data. For LibriSpeech, the streaming encoder used a 20-layer Emformer [35] with a 160 ms segment size, 512 attention dimensions, 2048 feed-forward dimensions, and 8 heads (63M parameters). With the same attention configurations, a 30-layer streaming Conformer (190M) [36] was used for MLS data, in which a chunk-based mask was implemented with a 320 ms average latency. Model input used 80 d filterbank features with a 10 ms frame rate were used with 1/4 down-sampling.

The standard RNN-T model and FT model from [8] were built to compare with Transducer-Llama. All of these used the same encoder. The standard RNN-T had a predictor consisting of two LSTM layers with 2048 hidden dimensions. With the same structure, the non-blank predictor of the FT baseline was pre-trained on text-only data and kept fixed during ASR training. The values of α and β in Eq. 1 were set to 0.6 following [8]. Aside from the non-blank predictor, Transducer-Llama shared the same settings as the FT baseline. During training, an embedding layer was used as the non-blank predictor. Aside from the LSTM LM used by the FT baseline, Llama2-0.5b [11] and Llama3-8b [12] were fine-tuned on the same text data using the proposed vocabulary adaptation, in

TABLE I
WER ON LIBRISPEECH TEST SETS FOR STREAMING MODELS (160MS SEGMENT SIZE). FFT LLAMA2 IS FULLY FINE-TUNED (FFT) LLAMA2.

Online Models	Test-clean	Test-other
RNN-T	3.65	8.96
Factorized Transducer [8]	2.97	7.77
Transducer-Llama		
w/ LSTM LM non-blank predictor	2.86	7.33
w/ Llama2 non-blank predictor	2.76	7.07
w/ FFT Llama2 non-blank predictor	2.54	6.59
w/ Llama3 non-blank predictor	2.47	6.53

TABLE II
ABLATION STUDIES ON THE WEAK-TO-STRONG LM SWAP METHOD USING LIBRISPEECH DATA.

Train-time Non-blank Predictor	Test-time Non-blank Predictor	MWER	Test-clean	Test-other
LSTM LM	LSTM LM	✗	3.11	7.83
LSTM LM	LSTM LM	✓	2.97	7.77
Stateless	LSTM LM	✗	3.20	7.59
Stateless	LSTM LM	✓	2.86	7.33
LSTM LM	Llama2	✗	2.95	7.45
LSTM LM	Llama2	✓	2.82	7.41
Stateless	Llama2	✗	3.39	7.47
Stateless	Llama2	✓	2.76	7.07
Llama2	Llama2	✗	3.03	7.63
Llama2	Llama2	✓	2.90	7.41
LSTM LM	FFT Llama2	✗	2.67	6.95
LSTM LM	FFT Llama2	✓	2.61	6.94
Stateless	FFT Llama2	✗	3.00	6.91
Stateless	FFT Llama2	✓	2.54	6.59
FFT Llama2	FFT Llama2	✗	2.86	7.51
FFT Llama2	FFT Llama2	✓	2.72	7.25

which the LLM Transformer layers are fixed. This paper also evaluated the case when the Llama2-0.5b Transformer layers were also updated, denoted as fully fine-tuned Llama2 (FFT Llama2). The LLM replaced the used LM of the Transducer-Llama as the non-blank predictor after RNN loss training.

For LibriSpeech, the ASR models were trained for 40 epochs, while for MLS, the ASR models were trained for 200k steps with a 5.3m total batch size. MWER training ran for up to a few thousand steps. The LM was trained for up to 40 epochs, while the vocabulary adaptation training for LLMs often stopped early after a few epochs. At decoding, the beam search size was 10.

C. Experimental Results

The proposed Transducer-Llama was evaluated on both LibriSpeech and MLS data. Ablation studies were conducted to verify the effectiveness of the vocabulary adaptation technique and the weak-to-strong LM swap.

1) *LibriSpeech Main Results:* As shown in Table I, the FT baseline outperformed the RNN-T baseline, which is consistent with [8]. In addition, the proposed Transducer-Llama approach provides a framework that fully leverages

TABLE III

ABLATION STUDIES ON VOCABULARY ADAPTATION USING LIBRISPEECH WITH LLAMA3 AS THE NON-BLANK PREDICTOR. WEAK-TO-STRONG LM SWAP AND MWER TRAINING WERE NOT USED. LLAMA3 TOKENIZE HAS 128K VOCABULARY SIZE COMPARED TO 5K FOR ASR.

Online Models	Tokenizer	Train Speed	Test	
			clean	other
Transducer-Llama	Llama3	1	3.02	7.44
Transducer-Llama	ASR	$\times 8.05$	2.76	7.36

TABLE IV

WER ON MLS TEST SETS FOR DIFFERENT MODELS. THE LAST COLUMN SHOWS THE AVERAGE WER ON THE 4 LANGUAGES. LSTM PREDICTOR HAS 20M PARAMETERS, LLAMA2 HAS 0.5B PARAMETERS, LLAMA3 HAS 8B PARAMETERS. OUR BUILT MODELS ARE MULTI-LINGUAL ASR.

Online Models	en	fr	it	nl	Avg
Factorized Transducer [8]	8.19	7.33	14.02	13.37	10.73
Transducer-Llama					
w/ LSTM LM non-blank predictor	8.26	6.28	12.30	12.39	9.80
w/ FFT Llama2 non-blank predictor	7.57	5.80	11.59	12.17	9.28
w/ Llama3 non-blank predictor	7.35	5.79	11.76	12.05	9.24
Offline Mono-lingual CTC w/ LM [22]	5.9	5.6	10.5	12.0	8.50
Offline Speech LLM [14]	6.2	5.5	10.8	11.3	8.45
Offline Transducer-Llama					
w/ FFT Llama2 non-blank predictor	5.76	5.00	10.38	10.28	7.86
w/ Llama3 non-blank predictor	5.59	4.96	10.43	10.22	7.80

LMs to enhance ASR performance. As more powerful LMs are used, the WER of the LM-Transducer continues to decrease. Compared to the strong FT baseline, up to 16.8% and 16.0% relative WER reduction (WERR) were achieved on test-clean and test-other, respectively. This also shows that with our vocabulary adaptation, the LLM can efficiently adapt to the ASR vocabulary while maintaining strong performance, offering advantages for practical ASR deployments. In addition, Llama3 slightly outperforms the fully fine-tuned (FFT) Llama2, but by keeping the Transformer layers fixed, it retains the potential for a broader range of downstream tasks. Moreover, the ASR improvement (6% WERR) of the Transducer-Llama with the LSTM LM as a non-blank predictor, compared to the FT baseline, highlights the advantages of the weak-to-strong LM swap strategy. Detailed ablation studies are given in Sec. IV-C2.

2) *Ablation Studies on Weak-to-Strong LM Swap:* As shown in Table II, even with the ILM fusion strategy (Eq. 1), integrating stronger LLMs during RNN-T loss training yields only minor ASR improvements. For example, compared to using an LSTM LM as the non-blank predictor, employing stronger fully fine-tuned (FFT) Llama2 at training results in only 4% WERR on test-other. However, when the weak-to-strong LM swap strategy is used, e.g. using an LSTM LM during training and a FFT Llama2 during decoding, the ASR WERR greatly increased and gave a 14% WERR on test-clean and 11% WERR and test-other. While using a stateless non-blank predictor during RNN-T loss training performs slightly worse than when using an LSTM LM after swapping to an LLM on test-clean, it achieves the best results after MWER

training. The use of a stateless predictor, being even weaker than an LSTM, has a larger gap to the LLM, so after swap, it does not surpass the performance of LSTM-trained models. MWER training is designed to address this issue and optimizes the LLM predictor integration, which is especially effective for the stateless non-blank predictor training case (up to 18.6% WERR). Using a weak LM as the non-blank predictor during RNN-T loss training speeds up the training process and prevents the model from relying on the accurate predictions of a strong LM, ensuring the acoustic encoder is properly trained.

3) *Ablation Studies on Vocabulary Adaptation:* Table III compares the Transducer-Llama performance using the ASR vocabulary versus its original Llama3 tokenizer. The output of the neural transducer is known to be memory-intensive because it is a 4-dimensional tensor, including the vocabulary dimension. Given the vocabulary size of Llama3 is much larger than that of the ASR system (about 26 times larger), this causes higher memory consumption for the neural transducer model and slows down the training speed. Moreover, using the smaller vocabulary size and the tokenizer trained from ASR data provide performance benefits, with 8.6% WERR on test-clean. Therefore, the vocabulary adaptation allows the LLM to be more flexibly integrated into speech systems.

4) *Multi-lingual LibriSpeech (MLS) results:* Experiments were also conducted on the large-scale MLS data, and the conclusions are generally consistent with those on LibriSpeech: Transducer-Llama can still fully utilize the LLMs on online multilingual ASR, and the weak-to-strong LM swap approach is effective. For example, Transducer-Llama outperformed the strong FT baseline with 8.7% WERR when the LSTM LM was used as the non-blank predictor. When the Llama3 was used, 13.8% WERR was achieved. Llama3 performs slightly better than FFT Llama2 on average, while its fixed Transformer layers preserve the potential for other tasks. This section further constructs an offline Transducer-Llama using the offline Conformer encoder (71M) from [14], in order to compare with other model architectures and published results. The proposed offline Transducer-llama demonstrates superior performance over monolingual + LM baselines (8.2% WERR), and outperforms an offline speech LLM [14] by 7.7% WERR.

V. CONCLUSIONS

This paper proposes the Transducer-Llama framework, which naturally integrates LLMs into online ASR. With the vocabulary adaptation technique, Transducer-Llama retains the flexibility of using the tokenizer specifically designed for ASR systems. By using the proposed weak-to-strong LM swap strategy, the LLM can be fully utilized to boost ASR performance. Moreover, with the vocabulary adaptation and a weak LM used during training, Transducer-Llama can maintain good training speed. In addition, MWER training further improves Transducer-Llama when using an LLM. Experiments on LibriSpeech and Multi-lingual LibriSpeech (MLS) data show that the proposed Transducer-Llama gave a 17% relative WER reduction (WERR) over a strong FT baseline and 32% WERR over an RNN-T baseline.

REFERENCES

- [1] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in *Proc. ICML*, 2006.
- [2] A. Graves, “Sequence transduction with recurrent neural networks,” *ArXiv*, vol. abs/1211.3711, 2012.
- [3] K. Deng and P. C. Woodland, “Label-synchronous neural transducer for adaptable online E2E speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 3507–3516, 2024.
- [4] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, “RNN-transducer with stateless prediction network,” in *Proc. ICASSP*, 2020.
- [5] X. Chen, Z. Meng, S. Parthasarathy, and J. Li, “Factorized neural transducer for efficient language model adaptation,” in *Proc. ICASSP*, 2022.
- [6] K. Deng and P. C. Woodland, “Decoupled structure for improved adaptability of end-to-end models,” *Speech Communication*, vol. 163, p. 103109, 2024.
- [7] Z. Meng, T. Chen, R. Prabhavalkar, Y. Zhang, G. Wang, K. Audhkhasi, J. Emond, T. Strohman, B. Ramabhadran, W. R. Huang, et al., “Modular hybrid autoregressive transducer,” in *Proc. SLT*, 2023.
- [8] J. Guo, N. Moritz, Y. Ma, F. Seide, C. Wu, J. Mahadeokar, O. Kalinli, C. Fuegen, and M. Seltzer, “Effective internal language model training and fusion for factorized transducer model,” in *Proc. ICASSP*, 2024.
- [9] R. Zhao, J. Xue, P. Parthasarathy, V. Miljanic, and J. Li, “Fast and accurate factorized neural transducer for text adaption of end-to-end speech recognition models,” in *Proc. ICASSP*, 2023.
- [10] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., “Language models are few-shot learners,” in *Proc. NeurIPS*, 2020.
- [11] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., “LLaMa 2: Open foundation and fine-tuned chat models,” 2023.
- [12] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, et al., “The Llama 3 herd of models,” *arXiv preprint arXiv:2407.21783*, 2024.
- [13] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al., “Training language models to follow instructions with human feedback,” *Proc. NeurIPS*, 2022.
- [14] Y. Fathullah, C. Wu, E. Lakomkin, J. Jia, Y. Shangguan, K. Li, J. Guo, W. Xiong, J. Mahadeokar, O. Kalinli, C. Fuegen, and M. Seltzer, “Prompting large language models with speech recognition abilities,” in *Proc. ICASSP*, 2024.
- [15] K. Deng, G. Sun, and P. C. Woodland, “Wav2Prompt: End-to-end speech prompt generation and tuning for llm in zero and few-shot learning,” *arXiv preprint arXiv:2406.00522*, 2024.
- [16] Z. Chen, H. Huang, O. Hrinchuk, K. C. Puvvada, N. R. Koluguri, P. Želasko, J. Balam, and B. Ginsburg, “BESTOW: Efficient and streamable speech language model with the best of two worlds in GPT and T5,” *arXiv preprint arXiv:2406.19954*, 2024.
- [17] B. Minixhofer, E. M. Ponti, and I. Vulić, “Zero-shot tokenizer transfer,” *arXiv preprint arXiv:2405.07883*, 2024.
- [18] K. Deng, Z. Yang, S. Watanabe, Y. Higuchi, G. Cheng, and P. Zhang, “Improving non-autoregressive end-to-end speech recognition with pre-trained acoustic and language models,” in *Proc. ICASSP*, 2022.
- [19] G. Dagan, G. Synnaeve, and B. Rozière, “Getting the most out of your tokenizer for pre-training and domain adaptation,” *arXiv preprint arXiv:2402.01035*, 2024.
- [20] J. Guo, G. Tiwari, J. Dropped, M. V. Segbroeck, C.-W. Huang, A. Stolcke, and R. Maas, “Efficient minimum word error rate training of RNN-transducer for end-to-end speech recognition,” in *Proc. Interspeech*, 2020.
- [21] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “LibriSpeech: an ASR corpus based on public domain audio books,” in *Proc. ICASSP*, 2015.
- [22] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, “MLS: A large-scale multilingual dataset for speech research,” in *Proc. Interspeech*, 2020.
- [23] F. Seide, M. Doulaty, Y. Shi, Y. Gaur, J. Jia, and C. Wu, “Speech RealLM – real-time streaming speech recognition with multimodal llms by teaching the flow of time,” *arXiv preprint arXiv:2406.09569*, 2024.
- [24] E. Tsunoo, H. Futami, Y. Kashiwagi, S. Arora, and S. Watanabe, “Decoder-only architecture for streaming end-to-end speech recognition,” *arXiv preprint arXiv:2406.16107*, 2024.
- [25] X. Ma, J. M. Pino, and P. Koehn, “SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation,” in *Proc. ACL/IJCNLP*, 2020.
- [26] K. Deng and P. C. Woodland, “Label-synchronous neural transducer for E2E simultaneous speech translation,” in *Proc. ACL*, 2024.
- [27] K. Dobler and G. de Melo, “FOCUS: effective embedding initialization for monolingual specialization of multilingual models,” in *Proc. EMNLP*, 2023.
- [28] L. Gee, A. Zugarini, L. Rigutini, and P. Torroni, “Fast vocabulary transfer for language model compression,” in *Proc. EMNLP (Industry Track)*, 2022.
- [29] B. Minixhofer, F. Paischer, and N. Rekabsaz, “WECHSEL: effective initialization of subword embeddings for cross-lingual transfer of monolingual language models,” in *Proc. NAACL-HLT*, 2022.
- [30] P.-Y. Chen, “Model reprogramming: Resource-efficient cross-domain machine learning,” in *Proc. AAAI*, 2024.
- [31] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, “Sequence-discriminative training of deep neural networks,” in *Proc. Interspeech*, 2013.
- [32] D. Povey and P. Woodland, “Minimum phone error and I-smoothing for improved discriminative training,” in *Proc. ICASSP*, 2002.
- [33] R. Schlüter, B. Müller, F. Wessel, and H. Ney, “Interdependence of language models and discriminative training,” in *Proc. ASRU*, 1999.
- [34] C. Shan, C. Weng, G. Wang, D. Su, M. Luo, D. Yu, and L. Xie, “Component fusion: Learning replaceable language model component for end-to-end speech recognition system,” in *Proc. ICASSP*, 2019.
- [35] Y. Shi, Y. Wang, C. Wu, C.-F. Yeh, J. Chan, F. Zhang, D. Le, and M. Seltzer, “Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition,” in *Proc. ICASSP*, 2021.
- [36] 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.