



Personalization for BERT-based Discriminative Speech Recognition Rescoring

Jari Kolehmainen¹, Yile Gu¹, Aditya Gourav¹, Prashanth Gurunath Shivakumar¹, Ankur Gandhe¹, Ariya Rastrow¹, and Ivan Bulyko¹

¹Amazon, US

jkolehm@amazon.com

Abstract

Recognition of personalized content remains a challenge in end-to-end speech recognition. We explore three novel approaches that use personalized content in a neural rescoring step to improve recognition: gazetteers, prompting, and a cross-attention based encoder-decoder model. We use internal de-identified en-US data from interactions with a virtual voice assistant supplemented with personalized named entities to compare these approaches.

On a test set with personalized named entities, we show that each of these approaches improves word error rate by over 10%, against a neural rescoring baseline. We also show that on this test set, natural language prompts can improve word error rate by 7% without any training and with a marginal loss in generalization. Overall, gazetteers were found to perform the best with a 10% improvement in word error rate (WER), while also improving WER on a general test set by 1%.

Index Terms: speech recognition, rescoring, personalization, prompting, gazetteers.

1. Introduction

Production automatic speech recognition (ASR) systems often utilize two passes, where a first pass generates a list of hypotheses, and a second pass rescores the hypotheses to identify likely transcriptions [1, 2, 3]. For each hypothesis, the second pass models calculate a score (typically, negative log-likelihood) that is combined with the first pass scores to determine the final score [4, 5]. Scores can be computed using a language model [4, 5, 6], a masked language model [7, 8], or other hypothesis-level models [9, 10]. Rescoring performance has been improved by using contextual information in recent literature. This includes audio [5, 11], lattice information from the first pass [6, 12], or personalized context [13, 14].

Common neural rescoring models [15, 5, 7, 9, 10] face challenges with personalized content such as personalized named entities or personal playlists as they cannot adapt to the user's preferences. In particular, this issue manifests itself for homophone names where neither semantic nor acoustic context can distinguish the correct transcription. Personalized and supplemental n-gram models can address these challenges, but they usually lack generalization and overfit to a specific use case, reducing their effectiveness [14]. Domain classifiers and other auxiliary models can be used to improve generalization properties. This, however, results in a more complex second pass architecture and higher maintenance costs [16]. Including textual context directly in a neural model has been shown effective in many tasks such as document classification [17], language modeling [18], and acoustic modeling [19]. In this study, we

demonstrate how personalized named entities can be incorporated into a neural rescoring pass.

The literature provides a wide variety of neural modeling approaches that utilize both unstructured and structured contextual information: techniques that use encoded context via a cross-attention framework [17, 20, 19]; retrieval-based approaches [18, 21]; or approaches that modify the model input via embedding augmentation [22, 23, 24]. For the last category, it can either be achieved by augmenting token-level input embeddings [22, 24] with *gazetteers*. Alternatively, additional context tokens can be processed as part of the input sequence following certain logic [23, 25, 26, 27]. The latter approach is referred to as *prompting*. Prompt embeddings can be trainable [23, 25] or discrete optimization can be performed on the existing embeddings to find optimal prompts, without the need to apply conventional model training [26].

There are auto-regressive rescoring models [4, 5, 6] that can be used for rescoring, but bidirectional discriminative models have recently shown superior performance [9, 10]. In this paper, we use a bidirectional transformer model, called RescoreBERT, as the baseline [10]. RescoreBERT is a rescoring model that predicts a single score for each hypothesis and is trained with a discriminative ASR loss, which directly minimizes the word error rate. The model architecture is summarized in Section 2.1 and illustrated in Fig. 1a.

To investigate personalization in neural rescoring, we use internal en-US data supplemented with each user's personalized named entities. We focus on three novel rescoring approaches: 1) gazetteers where we add trainable slot embeddings to the vanilla RescoreBERT model; 2) natural language prompting where we include prompts to contextualize the RescoreBERT model; and 3) an encoder-decoder architecture with cross-attention to personalized named entities. We show results for these models in different data scenarios, model sizes, and test sets. This study intends to demonstrate the level of improvement each approach can yield in recognizing personalized content and entities without compromising general recognition performance.

2. Proposed Approaches

2.1. Rescoring Model

We use the recently proposed RescoreBERT as the baseline rescoring model [10]. RescoreBERT model uses a pre-trained BERT model with a feed-forward network attached to the classification (CLS) token embedding to predict a single score for each hypothesis (see Fig. 1a). Unlike auto-regressive language models, RescoreBERT scores do not follow a proper probability distribution. In more formal terms, the final score is calculated

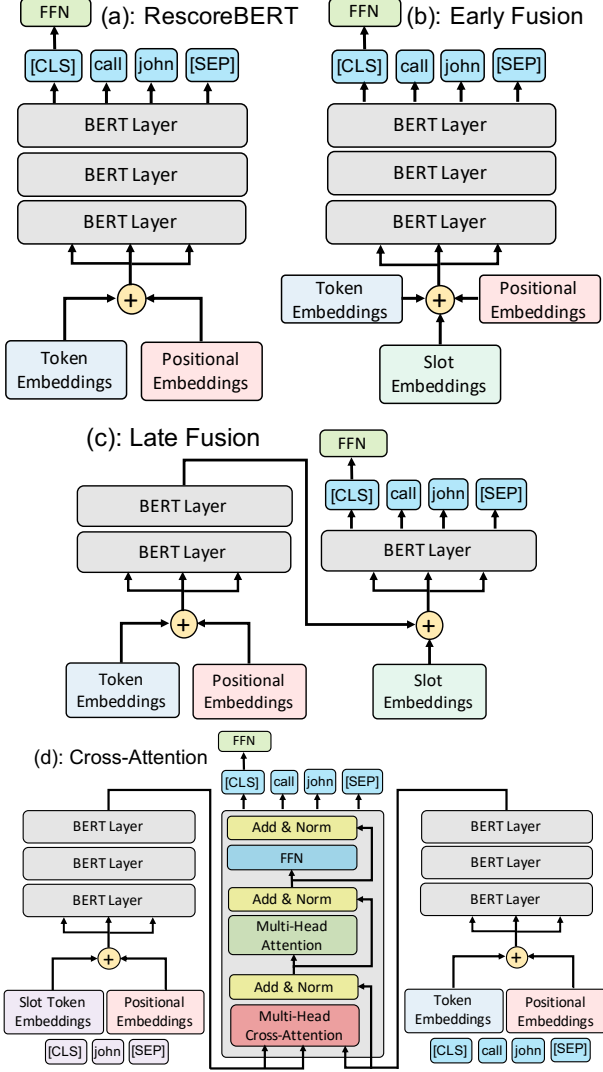


Figure 1: *Illustration of model architectures. (a) RescoreBERT model that attaches a feed-forward network at the classification token embedding of the BERT model. (b) Early fusion gazetteer that adds slot embeddings at the input side of the RescoreBERT model. (c) Late fusion gazetteer that adds the slot embeddings to the inputs of the last RescoreBERT model layer. (d) Cross-attention based encoder-decoder with shared encoders.*

by,

$$v_i = \alpha u_i + \beta s_i \quad (1)$$

where u_i is the first pass score; α and β are hyper-parameters (in this study they are 20 and 1, respectively); and s_i is the rescoring model score. The rescoring model score is computed by

$$s_i = f \circ g(E_t(x_i) + E_p). \quad (2)$$

Where f represents additional feed-forward layers operating on the CLS token embedding; $g = g_n \circ \dots \circ g_1$ is a transformer encoder consisting of n layers; $E_t(x_i)$ and E_p are token and (learnable) positional embeddings, respectively; and x_i are hypothesis tokens. In this study, the feed-forward-network f is a linear layer without any activation functions.

The rescoring model is trained using a standard minimum word error rate (MWER) loss calculated on n-best hypotheses from the first-pass: loss [28, 29, 30]:

$$L = \sum_i (\epsilon_i - \bar{\epsilon}) p_i, \quad (3)$$

where ϵ_i is the edit distance of i^{th} hypothesis; $\bar{\epsilon}$ is the mean edit distance of all hypotheses originating from the same utterance; and p_i is the normalized posterior distribution of the i^{th} hypothesis in the n-best, given by the following softmax:

$$p_i = \frac{\exp(-v_i)}{\sum_k \exp(-v_k)}. \quad (4)$$

2.2. Prompts

We consider a setting where each utterance has a set of tokenized strings D representing personalized named entities. In the prompt approach, RescoreBERT input is concatenated with a natural language prompt if any sub-string (continuous subsequence of tokens) in the hypotheses matches an entity in the set D . We call this a *match condition*. For example, if [entity] is among the personalized contact names for a particular hypothesis “call [entity]”, then the *match condition* is met. In this study, we did not perform any prompt tuning or prompt engineering and instead used a simple phrase “as i need to contact [entity]” as the augmented prompt. When there are multiple matched entities within a hypothesis, “and” is used in the prompt to separate them.

2.3. Gazetteers

For gazetteers, we process the input tokens x_i by tagging each token that matches an element in D . Only full matches are tagged and partial matches are ignored. The result is a binary sequence y_i , in which zeros indicate a non-match and ones indicate a match. Note that y_i has the same length as the input tokens x_i . An embedding layer can convert the slot tags y_i into dense representations, with tagged tokens (i.e. $y_i = 1$) generating a learnable embedding and non-tagged tokens (i.e. $y_i = 0$) generating a fixed non-learnable zero embedding. We consider two gazetteer variants: early fusion and late fusion, which differ in how the slot embeddings are added to the model.

In early fusion, the slot embeddings are added before the first transformer layer into the token and positional embeddings (see Fig. 1b). The resulting score is computed by

$$s_i = f \circ g(E_t(x_i) + E_p + E_s(y_i)), \quad (5)$$

where E_s are the slot embeddings. Note that the slot embeddings are added to the existing embeddings instead of replacing them. This will allow model to learn to distinguish salient words (i.e. “will”) from actual names.

In late fusion, slot embeddings are added to the last transformer layer (see Fig. 1c), the score is computed using:

$$s_i = f \circ g_n(E_s + g_{n-1} \circ \dots \circ g_1(E_t + E_p)). \quad (6)$$

The slot embeddings E_s are zero when none of the tokens are tagged, and the score is the same as the baseline RescoreBERT in Eq. (1), for both fusion variants.

2.4. Cross-Attention Based Encoder-Decoder

The last personalization approach we considered combines prompting and gazetteers ideas using a cross-attention architecture, as illustrated in Fig. 1d. We use string-matching as

Table 1: *Used training data sources, testing and validation datasets.*

Dataset Name	Hours of Audio
Personalized named entities training set	131
General training set	91,656
Personalized validation set	40
Personalized test set	43
General test set	141

before to find all the matching entities from hypotheses that are in D . The matched tokenized personalized entities are concatenated with a classification token (CLS) and separated by separation token (SEP) to form slot token sequence z_i . Hence, z_i will appear as [CLS] [entity #1] [SEP] [entity #2] [SEP] ... [SEP]. The slot tokens z_i are then encoded using the same encoder as the hypothesis input tokens x_i . Both the hypothesis and the encoded slot embeddings are used as inputs to a non-causal transformer decoder layer to extract a classification token for the scoring feed-forward network. This is more formally expressed by

$$X_i = g(E_t(x_i) + E_p) \quad (7)$$

$$Z_i = g(E_t(z_i) + E_p) \quad (8)$$

$$s_i = f \circ m(Z_i, X_i), \quad (9)$$

where $m(\cdot, \cdot)$ is a decoder layer without a causal mask where the first entry represents the keys and values of the multi-head cross-attention and the second entry is for the query (see Fig. 1d). If there are no matching name, the z_i is a single classification token. In this study, the decoder layer used the same hyper-parameters as the encoder (e.g. number of attention heads, hidden size, etc.).

3. Experiments

3.1. Data

We use de-identified far-field en-US datasets gathered from interactions with a virtual voice assistant in this study. Training data is split into two parts: The first part consists of personalized utterances where the ground truth transcription contains a reference to a personalized named entity; a second part consists of randomly selected utterances from usage traffic that may not contain references to the personalized entities of users. For example, the personalized utterances could look like “alexa make a phone call to john doe”. These two data sources are combined to create the final training data set. Samples from the personalized data source are added until a predetermined “personalized” fraction is reached. Users typically have a few thousand records of personalized “contact name” entities, each with a first and last name.

3.2. Models

3.2.1. First Pass Model

We use a recurrent neural network transducer (RNN-T) based first pass model that also employs neural personalization [19] and is supplemented with a shallow fusion n-gram model. The first pass score consists of two parts:

$$u_i = \gamma \log P_{CA-RNN-T} + \epsilon \log P_{SF}, \quad (10)$$

where $P_{CA-RNNT}$ is the contextual-aware RNN-T model [19] probability; P_{SF} is the shallow fusion n-gram probability; γ and

ϵ are interpolation weights. The context-aware RNN-T can improve the word error rate upto $\sim 35\%$ and the shallow-fusion by additional $\sim 5\%$ compared with a standard RNN-T model [19].

3.2.2. Rescoring Models

We use two different rescoring BERT models in this study: (i) tiny BERT; and (ii) big BERT. Both BERT models use a sentence-piece tokenizer with a multilingual vocabulary of $\sim 153k$ tokens (trained on the MC4 dataset [31]). Roughly $\sim 30k$ of these tokens are from English language. The hyper-parameters for the model architecture are outlined in Table 2.

Table 2: *Summary of BERT model hyper-parameters.*

Attribute	Tiny BERT	Big BERT
Layer Parameters	$\sim 5M$	$\sim 170M$
Embedding Parameters	$\sim 49M$	$\sim 157M$
Hidden Size	320	1024
Number of Layers	4	16
Number of Attention Heads	16	16
Intermediate Layer Dimension	1200	3072
Dropout	0.1	0.1

Both BERT models were first pre-trained on the MC4 dataset [30], and then domain-adapted using internal in-domain data. The BERT models were used to initialize the baseline RescoreBERT models, followed by MWER-based training on the general training data [10]. For the proposed personalization approaches in Section 2, we initialize each model with the baseline RescoreBERT from the previous step.

3.3. Training and Evaluation Protocol

All the models used the public hugging-face implementation of BERT [32] (model parameters and tokenizer are different). DeepSpeed [33] was used for training and evaluating all models. Batch size was 256 and 128 for tiny and large BERT models, respectively. All experiments used a learning-rate decay policy and an Adam optimizer with default parameters. The initial learning rate was 10^{-5} for the tiny BERT experiments and 10^{-6} for the big BERT model experiments. An early stopping strategy, based on the personalized validation set, was applied (see Table 1).

4. Results and Discussion

Relative word error rate reduction (WERR) numbers using vanilla RescoreBERT models, compared to a baseline without a rescoring pass, are shown in Table 3. Using Oracle rescoring (e.g. picking the hypothesis with the smallest edit distance from truth) improves the first pass significantly, demonstrating the bound on WER improvements on both test sets.

RescoreBERT (see Fig. 1a) improves on the general test set, but degrades the personalized test set. This is likely due to the fact that the first pass is already context aware, while the RescoreBERT models are not. The larger RescoreBERT amplifies this characteristic by causing a more pronounced degradation on the personalized test set. A fine-tuned RescoreBERT model reduces degradation but fails to match the first-pass WER on the personalized test set.

By applying natural language prompts without any training to RescoreBERT models (see Table 4), WER improves significantly over the first pass with some degradation on the general test set. We hypothesize that this is particularly effective for the

Table 3: Baseline WERR results of rescoring models in the personalized and general test sets relative to the first pass (no rescoring).

Model	Personalized	General
Oracle	-57%	-58%
Tiny RescoreBERT	+3.9%	-5.3%
Big RescoreBERT	+4.8%	-7.1%
Tiny RescoreBERT (fine-tuned)	+2.5%	-5.3%
Big RescoreBERT (fine-tuned)	+1.7%	-6.8%

Table 4: Rescoring model WERR results in personalized and general test sets. Upper section: WERR relative to tiny RescoreBERT. Lower section: WERR relative to the big RescoreBERT. Negative sign corresponds to WER reduction and positive sign WER increase. Percentage sign in brackets in the model name denotes personalized training data fraction.

Model	Personalized	General
Untrained Prompt	-7.1%	0%
Frozen Late	-4.5%	0%
Frozen Early	-4.9%	0%
Trained Late	-13%	+3.9%
Trained Early	-13%	+3.8%
Trained C-A	-14%	+5.1%
Untrained Prompt	-12%	+3.0%
Frozen Early	-6.1%	0%
Trained Early (1%)	-12%	-0.7%
Fine-tuned Prompt (1%)	-13%	+0.4%

RescoreBERT model. The first-pass model penalizes longer utterances by assigning higher scores. Due to MWER training, the rescoring model learns to counter this by predicting a high-ranking score for longer utterances. By increasing the length of the hypothesis with adding semantically correct prompts, we believe that the hypothesis will receive better scores, thus getting ranked better than other hypotheses. We tested this anecdotally by trying out a variety of semantically correct and incorrect prompts and observed the same expected behavior.

Both early and late fusion gazetteers require additional training for the slot embeddings E_s in Eqs. (5)-(6). Freezing the weights of the vanilla RescoreBERT portion of the model, and training only the slot embeddings E_s (“Frozen model”), improves the personalized-entity WER, with minimal degradation on the general test set compared with the baseline RescoreBERT model (see Table 4). Early fusion yields a slightly larger improvement than the late fusion gazetteer. In spite of the training of the slot embeddings, the “frozen model” gazetteers perform worse than prompting, which does not require training.

Training rescoring models with personalized data significantly improved WER on the personalized test set, but also caused degradation on the general test set. In addition, early fusion produced slightly better results than late fusion, so for the remainder of the experiments, we focus on early fusion. Cross-attention based encoder-decoder model gave the largest improvement on the personalized data, but also caused the largest degradation on the general data. This is partly explained by the larger number of trainable parameters.

Fig. 2 shows training results for the models when general data is mixed into the training data. Using data mixing effectively mitigates the general test set degradation without significant loss of accuracy on the personalized test set with the

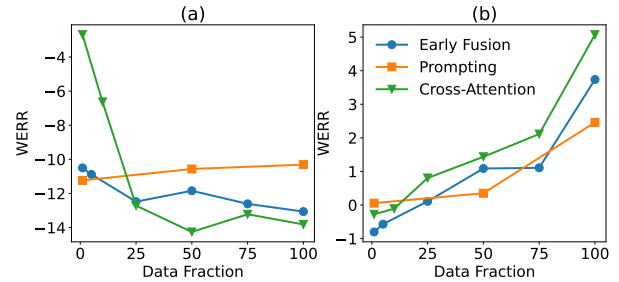


Figure 2: WERR respect to the tiny RescoreBERT model for early fusion gazetteer (blue line with circles), prompting (orange line with squares), and cross-attention (red line with triangles). Data fraction shown on the x axis is the fraction of personalized entity data in the used training data. Positive WERR denotes increase and minus sign reduction of WER respect to tiny RescoreBERT. (a) the personalized entity test set WERR for chosen approaches. (b) the general test set WERR for chosen approaches.

exception of the cross-attention approach. The cross-attention model WER on the personalized test set became significantly worse when no general test set degradation was allowed. In the personalized test set, early fusion gazetteer performed slightly worse than cross-attention, but was less sensitive to data fraction. The early fusion model was also the only one that improved WER on the general test set over the baseline RescoreBERT, while also achieving over 10% WERR on the personalized test set.

The bottom half of Table 4 presents the results of repeated experiments using the big BERT model. Prompting with the big RescoreBERT model proved very effective for personalization and provided a similar level of improvement as the trained models. However, it caused a larger degradation on the general test set than prompting with the tiny RescoreBERT. The trained early fusion gazetteer results improved in every aspect using a larger transformer.

5. Conclusions

We compared three different approaches to include personalized named entities in rescoring: (i) early and late fusion gazetteers; (ii) prompts; and (iii) cross-attention based encoder-decoder model. The effectiveness of these approaches was explored using internal data supplemented with personalized named entities. When trained with personalized data, all the approaches could deliver over 10% WERR improvement over the baseline RescoreBERT model for the personalized test set. However, this leads to degradation in the general test set. Mixing some fraction of general data did address this issue for prompting and gazetteers with small loss of performance in the personalization test set. Cross-attention was more sensitive to the data fraction and we did not find a data mixing ratio that would yield good performance in both test sets. We also found that prompting gave 7% improvement over the baseline WER without any training. This could be particularly useful for cases where training data is limited or not available. gazetteers proved to be the best all around approach as it was the only model that could give over 10% improvement in the personalization while improving the general data test set at the same time. Both the gazetteers and prompting approaches performance become better with larger model size suggesting that they are scalable.

6. References

- [1] Y. Xia, F. Tian, L. Wu, J. Lin, T. Qin, N. Yu, and T.-Y. Liu, "Deliberation networks: Sequence generation beyond one-pass decoding," *Advances in neural information processing systems*, vol. 30, 2017.
- [2] K. Hu, T. N. Sainath, R. Pang, and R. Prabhavalkar, "Deliberation model based two-pass end-to-end speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7799–7803.
- [3] K. Hu, R. Pang, T. N. Sainath, and T. Strohman, "Transformer based deliberation for two-pass speech recognition," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 68–74.
- [4] H. Huang and F. Peng, "An empirical study of efficient asr rescoring with transformers," *arXiv preprint arXiv:1910.11450*, 2019.
- [5] A. Gandhe and A. Rastrow, "Audio-attention discriminative language model for asr rescoring," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7944–7948.
- [6] P. Pandey, S. D. Torres, A. O. Bayer, A. Gandhe, and V. Leutnant, "Lattention: Lattice-attention in asr rescoring," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7877–7881.
- [7] J. Salazar, D. Liang, T. Q. Nguyen, and K. Kirchhoff, "Masked language model scoring," *arXiv preprint arXiv:1910.14659*, 2019.
- [8] G. Liu and L. Cao, "Code-switch speech rescoring with monolingual data," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6229–6233.
- [9] H. Futami, H. Inaguma, M. Mimura, S. Sakai, and T. Kawahara, "Asr rescoring and confidence estimation with electra," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 380–387.
- [10] L. Xu, Y. Gu, J. Kolehmainen, H. Khan, A. Gandhe, A. Rastrow, A. Stolcke, and I. Bulyko, "Rescorebert: Discriminative speech recognition rescoring with bert," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6117–6121.
- [11] S. Kim, K. Li, L. Kabela, R. Huang, J. Zhu, O. Kalinli, and D. Le, "Joint audio/text training for transformer rescorer of streaming speech recognition," *arXiv preprint arXiv:2211.00174*, 2022.
- [12] L. Dai, L. Chen, Z. Zhou, and K. Yu, "Latticebart: Lattice-to-lattice pre-training for speech recognition," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6112–6116.
- [13] D. Filimonov, R. T. Gadde, and A. Rastrow, "Neural composition: Learning to generate from multiple models," *arXiv preprint arXiv:2007.16013*, 2020.
- [14] A. Gourav, L. Liu, A. Gandhe, Y. Gu, G. Lan, X. Huang, S. Kalmane, G. Tiwari, D. Filimonov, A. Rastrow *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)*. IEEE, 2021, pp. 7348–7352.
- [15] A. Raju, D. Filimonov, G. Tiwari, G. Lan, and A. Rastrow, "Scalable multi corpora neural language models for asr," *arXiv preprint arXiv:1907.01677*, 2019.
- [16] L. Liu, Y. Gu, A. Gourav, A. Gandhe, S. Kalmane, D. Filimonov, A. Rastrow, and I. Bulyko, "Domain-aware neural language models for speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7373–7377.
- [17] F. Ruggeri, M. Lippi, and P. Torrioni, "Membert: Injecting unstructured knowledge into bert," *arXiv preprint arXiv:2110.00125*, 2021.
- [18] Z. Zhong, T. Lei, and D. Chen, "Training language models with memory augmentation," *arXiv preprint arXiv:2205.12674*, 2022.
- [19] K. M. Sathyendra, T. Muniyappa, F.-J. Chang, J. Liu, J. Su, G. P. Strimel, A. Mouchtaris, and S. Kunzmann, "Contextual adapters for personalized speech recognition in neural transducers," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8537–8541.
- [20] F.-J. Chang, J. Liu, M. Radfar, A. Mouchtaris, M. Omologo, A. Rastrow, and S. Kunzmann, "Context-aware transformer transducer for speech recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 503–510.
- [21] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. B. Van Den Driessche, J.-B. Lespiau, B. Damoc, A. Clark *et al.*, "Improving language models by retrieving from trillions of tokens," in *International conference on machine learning*. PMLR, 2022, pp. 2206–2240.
- [22] B. Fetahu, A. Fang, O. Rokhlenko, and S. Malmasi, "Dynamic gazetteer integration in multilingual models for cross-lingual and cross-domain named entity recognition," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 2777–2790.
- [23] K. Ganesan, P. Bamdev, A. Venugopal, A. Tushar *et al.*, "N-best asr transformer: Enhancing slu performance using multiple asr hypotheses," *arXiv preprint arXiv:2106.06519*, 2021.
- [24] X. Hu, Z. Zhou, Y. Sun, J. Kersten, F. Klan, H. Fan, and M. Wiegmann, "Gazpne2: A general place name extractor for microblogs fusing gazetteers and pretrained transformer models," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16 259–16 271, 2022.
- [25] Y. Wen, N. Jain, J. Kirchenbauer, M. Goldblum, J. Geiping, and T. Goldstein, "Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery," *arXiv preprint arXiv:2302.03668*, 2023.
- [26] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh, "Autoprompt: Eliciting knowledge from language models with automatically generated prompts," *arXiv preprint arXiv:2010.15980*, 2020.
- [27] M. Nie, M. Yan, C. Gong, and D. Chuxing, "Prompt-based re-ranking language model for asr," *Proc. Interspeech 2022*, pp. 3864–3868, 2022.
- [28] T. Hori, C. Hori, S. Watanabe, and J. R. Hershey, "Minimum word error training of long short-term memory recurrent neural network language models for speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 5990–5994.
- [29] R. Prabhavalkar, T. N. Sainath, Y. Wu, P. Nguyen, Z. Chen, C.-C. Chiu, and A. Kannan, "Minimum word error rate training for attention-based sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4839–4843.
- [30] W. Wang, T. Chen, T. N. Sainath, E. Varianni, R. Prabhavalkar, R. Huang, B. Ramabhadran, N. Gaur, S. Mavandadi, C. Peyser *et al.*, "Improving rare word recognition with lm-aware mwer training," *arXiv preprint arXiv:2204.07553*, 2022.
- [31] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv e-prints*, 2019.
- [32] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: <http://arxiv.org/abs/1810.04805>
- [33] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He, "Zero: Memory optimizations toward training trillion parameter models," in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2020, pp. 1–16.