DOC-RAG: ASR Language Model Personalization with Domain-Distributed Co-occurrence Retrieval Augmentation

Puneet Mathur[*‡], **Zhe Liu**[†], **Ke Li**[†], **Yingyi Ma**[†], **Gil Keren**[†], **Zeeshan Ahmed**[†],
Dinesh Manocha[‡], **Xuedong Zhang**[†]
‡ University of Maryland, College Park
‡ {puneetm,dmanocha}@umd.edu
† Meta
† {zheliu,kli26,yingyima,gilkeren,ahzee,xuedong}@meta.com

Abstract

We propose DOC-RAG - `Domain-distributed Co-occurrence Retrieval Augmentation` for ASR language model personalization aiming to improve the automatic speech recognition of rare word patterns in unseen domains. Our approach involves contrastively training a document retrieval module to rank external knowledge domains based on their semantic similarity with respect to the input query. We further use n-gram co-occurrence distribution to recognize rare word patterns associated with specific domains. We aggregate the next word probability distribution based on the relative importance of different domains. Extensive experiments on three user-specific speech-to-text tasks for meetings, TED talks, and financial earnings calls show that DOC-RAG significantly outperforms strong baselines with an 8-15% improvement in terms of perplexity and a 4-7% reduction in terms of Word Error Rates in various settings.

Keywords: language modeling, retrieval augmentation, LM personalization, speech recognition

1. Introduction

Language modeling is a core problem in natural language processing and is critical for automatic speech recognition (ASR) (Mikolov et al., 2010; Chen et al., 2015; Xu et al., 2018). Recently, Transformer-based LMs trained on large corpora have been extensively used for next-word prediction tasks and in the re-scoring stage of ASR systems (Irie et al., 2019a; Li et al., 2020a). Language models tend to memorize knowledge within their parameters during their training process (Petroni et al., 2019; Jang et al., 2022). The existence of user-preferred word patterns, named entities, and other domain-specific tail words that are not seen frequently in the training data make it difficult to personalize LMs for ASR second-pass re-scoring for unseen users and domains(Schick and Schütze, 2019; Maynez et al., 2020; Serai et al., 2022).

Retrieval augmentation (Lewis et al., 2020) has been recently proposed to adapt LMs to external world knowledge at inference time by using a retrieval mechanism to select and attend over relevant knowledge from an external data store to help inform its predictions(Naik et al., 2022; Liu et al., 2022; Borgeaud et al., 2022). Prior research has explored explicit memorization through k-Nearest Neighbor Language Models (kNN-LM) (Khandelwal et al., 2020), attention-based history through Grave et al., and non-parametric retrieval-based LM pre-training such as REALM (Guu et al., 2020) and RAG (Lewis et al., 2020). However, these methods were initially proposed to enhance the LM memorization capabilities rather than personalizing LMs to specific domains or users.

Our work drives motivation from the hypothesis that rare word patterns are domain/user-specific. By augmenting LM predictions with n-gram probabilities from a subset of query-relevant users/domains may address the problem of ASR LM personalization. To personalize ASR models without the need to continually retrain LMs for newer information, we propose - `Domain-Distributed Co-occurrence` (DOC-RAG), a novel retrieval augmentation approach that augments a pre-trained language model with a knowledge retriever which is trained via contrastive learning to rank textual documents/recordings from an external knowledge data store based on their semantic similarity with the input query. Our approach rewards retrievals that are contextually relevant to the input query while penalizing uninformative retrievals by assigning a probability distribution over the external knowledge domains to appropriately weigh their individual contribution.

Inspired by (Mathur et al., 2023), we address the challenge of capturing personalized word patterns associated with specific users/domains by exploiting bi-gram word frequencies from a subset of highly related and overlapping domains to the input query. We aggregate the target word probability distribution from different domains, weighted according to their relative importance to the query for the next word prediction and ASR second-pass re-scoring. The main contributions of this work are:

- We propose DOC-RAG, a

[*] Work done during internship at Meta

LREC-COLING 2024, pages 5132–5139
20-25 May, 2024.

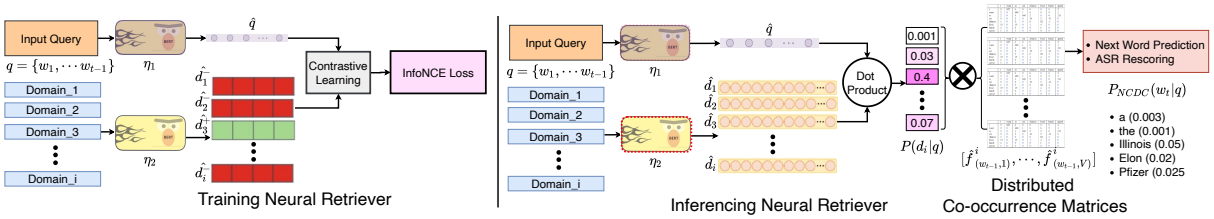

Figure 1: Domain-distributed Co-occurrence Retrieval Augmentation: (Left) Input query q and domain-specific text/recordings are encoded using BERT models (η_1, η_2), which are used as an input to the Neural Retriever and trained via contrastive learning to score correct positive pairs higher than negative query–context pairs. (Right) At inference, we compute the relevance score $P(d_i|q)$ between the query and each domain d_i as a dot product of their encoded representations using pre-trained BERT models (see red). Distributed co-occurrence matrices for each target domain represent the bi-gram frequencies $\hat{f}^i_{(w_{t-1},w_t)}$. We sum the word co-occurrence probabilities weighted by the relevance of the selected domain to obtain the probability $P(w_t|q)$ for next-word prediction.

`Domain-distributed Co-occurrence Retrieval Augmentation` that contrastively trains a neural retriever to select the most relevant external knowledge domains to the input query and uses n-gram word co-occurrence distribution to bias LM predictions.

- Experimental results show that our proposed approach achieves SOTA results on four datasets: WikiText-103, Earnings conference calls, AMI Meetings Corpus, and TED LIUMv3, significantly outperforming prior systems by a reduction of 8-15% in perplexity and 4-7% in Word Error Rate (WER).

2. Methodology

Fig. 1 describes our proposed DOC-RAG that biases the next word predictions from a base LM (pre-trained on a generic corpus G) based on the relevance of K unseen domains/users $d_1, d_2, \cdots, d_K$ to the input query $q = \{w_1, w_2, \cdots, w_{t-1}\}$. We hypothesize that rare words are domain-specific and that their distribution varies with topics/users. Fig. 1-Left illustrates the contrastively trained retriever of DOC-RAG which selects the best matching domains/users from a large external knowledge base for the input query. Fig. 1-Right shows how DOC-RAG augments the next word probability distribution $P_{LM}(w_t|q)$ over the target token w_t from the LM with domain-distributed word co-occurrence information. DOC-RAG calculates the probability distribution of NWP over the vocabulary by conditioning on the relevance of the underlying domains to the incoming query (q) based on the retrieved out-of-domain training set as $P_{DOC-RAG}(w_t|q) = \sum_{i=1}^{K} P(d_i|q) \times P(w_t|d_i, q)$. The query in the experiments can refer to either a partial text string for which we aim to find the next word (Next Word Prediction Task) or it may refer to the n-best hypothesis from audio models that are rescored based on language model perplexity (ASR 2nd-pass rescoring).

Encoding Query and Domain-specific Training Corpus: Given a query q and a large number of target domain contexts (d_i), we map them to fixed-length vectors using separate encoders η_1 and η_2, respectively. We use BERT (Devlin et al., 2019), a bidirectional Transformer architecture to encode the query and the domain context using the [CLS] token representation from the last layer. The query embedding is represented by $\hat{q} = \eta_1(q)$ and the domain context is represented by $\hat{d}_i = \eta_2(d_i)$. If the input length is larger than 512, the context embedding is calculated as the average of all sentence embedding c_j as $\hat{d}_i = \eta_2(d_i) = \sum_{j=0}^{l} \eta_2(c_j)$.

Training DOC-RAG Retriever: The neural retriever inputs the set of input query and domain contexts to output a relevance score $s(q, d_i)$ for each domain d_i. We use the dot product between the encoded query and the context vectors as the scoring function $s(q, d_i) = \langle \hat{q}, \hat{d}_i \rangle$. During training, we use contrastive learning to teach the model to discriminate and to score positive pairs (from the same domains) higher than negative (from different domains) query–context pairs. Formally, given a query q with an associated positive domain d^+, and a pool of negative domains (d_i^-), the contrastive InfoNCE loss compares the positive and the negative pairs based on the relevance score as defined below with τ as a temperature parameter.

$$L(q, d) = \frac{-\exp(s(q, d^+)/\tau)}{\exp(s(q, d^+)/\tau) + \sum_{i=1}^{K} \exp(s(q, d_i^-)/\tau)}$$

Retrieving Relevant Domains: At inference, the probability of the retriever model choosing a particular domain from the out-of-domain training set $P(d_i|q)$ is computed as the relevance score between the query and the domain context as $P(d_i|q) = s(q, d_i)$ encoded by the trained retriever.

Constructing Distributed Co-occurrence Matrix: Words that occur together in the out-of-domain training set are more likely to trigger together at inference as well. This chance co-occurrence is also highly dependent on the underlying domain

of the query. Hence, we construct word-level co-occurrence matrices corresponding to each target domain d_i. To represent possible next words, we compute the bi-gram frequencies $\hat{f}^i_{(w_{t-1},w_t)}$ for all vocabulary words V in a particular domain/user document d_i. For the last token in the input query sequence q, the probability of the target token for the next word prediction (NWP) conditioned on the selected domain is calculated as $P(w_t|d_i, q) = [\hat{f}^i_{(w_{t-1},1)}, \hat{f}^i_{(w_{t-1},2)}, \cdots, \hat{f}^i_{(w_{t-1},V)}]$.

Language Model Augmentation: DOC-RAG computes the retrieved next-word probability by summing the word co-occurrence probabilities weighted by the relevance of the selected domain to obtain the probability distribution of the next word prediction across each word as $P_{DOC-RAG}(w_t|q) = \sum_{i=0}^{K} s(q, d_i) \cdot [\hat{f}^i_{(w_{t-1},1)}, \cdots, \hat{f}^i_{(w_{t-1},V)}]$.

Next, we estimate the retrieved next-word probability distribution through DOC-RAG retrieval augmentation ($p_{DOC-RAG}$) with the language model output (P_{LM}) using a hyperparameter λ to produce the final NWP probability distribution as $P(w_t|q) = \lambda P_{DOC-RAG}(w_t|q) + (1 - \lambda)P_{LM}(w_t|q)$.

3. Experiments

Datasets: Inspired by (Mathur et al., 2023), we use Librispeech (Panayotov et al., 2015) text for LM pre-training. We evaluate LM personalization on two text datasets - WikiText-103 (Merity et al., 2017), financial earnings calls corpus (Earnings-21+22) (Rio et al., 2021; Del Rio et al., 2022); and two speech datasets - AMI Meetings (Kraaij et al., 2005), speaker TED talks (Hernandez et al., 2018). To study the personalization of ASR LMs, we reformulated existing datasets to identify explicit users/domains (wiki page, financial company, speaker, or meeting). For each dataset, we combined the original train/val/test portions and split user-based data in the ratio of 70:10:20 such that each user/domain appears only in one of the splits. Table 1 shows domain distribution and corpus size. Domains in our work refer to the categories with a distribution different from the data used to train the base model. It refers to different users/topics/call recordings based on the dataset. Domains in the AMI Meeting corpus are formed based on speaker IDs. Domains in Earnings-21+22 data correspond to different companies. A specific domain in the TED-LIUM v3 dataset refers to a particular user. Domains in Wikitext-103 correspond to individual Wikipedia pages.

Language Model Architecture: Inspired by (Mathur et al., 2023), we experiment with both LSTM and Transformer LMs. LSTM model configurations: 2 layers, 300-d embedding layer, hidden dimension of 1500. Transformer LM configurations: 4 layers encoder-decoder, 12 heads, 128-d hidden representations, feed-forward layer of 3072-d. We use a pre-trained RNN-T ASR Model with Emformer encoder (Shi et al., 2021), LSTM predictor, and a joiner with 80M parameters for generating ASR n-best hypotheses.

Retriever Model Architecture: We use asymmetric dual encoders for domain retrieval to overcome domain distribution shift. We leverage two different Bert models (Devlin et al., 2019) for query and context encoding. We use pre-trained BERT with frozen weights for the query encoder. The BERT model for context encoder is fine-tuned via contrastive learning.

Pre-training LMs: Inspired by (Mathur et al., 2023), LSTM and Transformer LMs are pre-trained on Librispeech (Panayotov et al., 2015) training set for 25 epochs with a batch size of 256, Adam optimizer and cross-entropy loss for NWP task and benchmarked on the least perplexity of the Librispeech validation set.

Adaptation to Unseen Domains: Inspired by (Mathur et al., 2023), we evaluate the retrieval augmentation Without fine-tuning (LM pre-trained on generic corpus) and with fine-tuning (LM pre-trained on generic corpus and fine-tuned on out-of-domain train corpus). Evaluation is benchmarked on the out-of-domain test set.

Baselines: Inspired by (Mathur et al., 2023), we benchmark the following baselines:

- **(i) LSTM/Transformer**: Language model without any augmentation
- **(ii) Neural Cache Model** (Grave et al.) LM augmented with a continuous cache memory of previous hidden states. The stored keys are used to retrieve the next word through a dot product-based memory lookup with the query.
- **(iii) kNN-LM** (Khandelwal et al., 2020): Following (Das et al., 2022), kNN-LM memorizes context vectors from out-of-domain train set in an external data store. At inference, LM output is interpolated with the k-nearest neighbors of the decoder output representations.
- **(iii) RAG** $w\backslash$ **DPR**: Retriever Augmented Generation (Lewis et al., 2020) with Dense Passage Retriever for ranking.

Ablation Studies: We run the following ablation studies:

- **(i)** DOC-RAG $w\backslash o$ **Contrastive Retriever**: query and domain context encoded through a pre-trained BERT to compute relevance scores. We evaluate frozen Bert models for both query and context encoding.

Dataset	Train	Val	Test	Vocab Size	ASR Application	# Domains
Earnings-21+22	49.6K	7.1K	14.2K	20K	Earning Call	169
AMI Meeting Corpus	17.1K	2.7K	5.8K	11K	Meeting Recording	135
TED-LIUM v3	188.9K	26.6K	9.3K	46K	TED Talk	2351
Wikitext-103	2M	300K	10K	200K	Wikipedia Page	30k

Table 1: Data Statistics

- **(ii) DOC-RAG $w\backslash$ Distributed Co-occurrence** : No retriever step, a single co-occurrence matrix computed over combined out-of-domain train set of all users/domains.

Evaluation: (1) Word-level perplexity scores to evaluate LM performance for next-word prediction. (2) Word Error Rate (WER) for ASR second-pass re-scoring in speech datasets. Results report minimal perplexity by iterating the interpolation parameter λ between $(0, 1)$ in increments of 0.1.

4. Results and Analysis

Perplexity Evaluation: Table 2 compares the perplexity of the proposed DOC-RAG retrieval augmentation against baselines. We observe that the Neural Cache model (Li et al., 2020b) is ineffective due to its inability to handle long-range dependencies compared to other baselines. kNN-LM (Khandelwal et al., 2020) decreases perplexity by 5-10%, yet faces difficulties due to the non-parametric fuzzy characteristic of k-nearest context spans within tens of millions of stored contexts throughout the entire data store. This leads to sub-optimal retrieval of contexts from domains unrelated to the input query. RAG (Lewis et al., 2020) is the strongest baseline but has the drawback of not explicitly capturing user-specific word patterns. Our proposed DOC-RAG achieves state-of-the-art performance as it improves the perplexity scores by a significant margin on WikiText-103 ($54.6 - 50.5\%$ w\o fine-tuning, $8.5 - 12.6\%$ with fine-tuning), Earnings21+22 ($37.4 - 37.7\%$ w\o fine-tuning, $8.4 - 9.2\%$ with fine-tuning), AMI Meeting Corpus ($61.2 - 61.9\%$ w\o fine-tuning, $5.3 - 9.2\%$ with fine-tuning), and TED LIUMv3 ($19.1 - 22.2\%$ w\o fine-tuning, $2.2 - 2.8\%$ with fine-tuning). These experiments prove that contextually matching queries with external domains via contrastive learning improves retrieval task performance and reinforces the NWP task. Further, Table 2 shows that our proposed approach improves WER by 2-5% for second-pass ASR rescoring on AMI Meetings and TED LIUMv3 datasets due to its ability to correctly recognize domain-specific rare words in n-best hypotheses produced by the audio model. Variations in perplexity WER scores for LSTM and Transformers are highly correlated with the domain of the training data. Overall, Transformers are better than LSTM for ASR personalization tasks due to a higher number of parameters.

Ablation Analysis: Replacing the distributed word co-occurrence with a unified bi-gram frequency for all external domains significantly deteorites LM performance across various settings. This shows the advantage of incorporating distributed word co-occurrences for exploiting domain/user-specific word patterns. Using a contrastive retriever in place of a Dense Passage Retriever further improves the performance as it is able to use the rare word patterns from different domains based on their contextual similarity to the input, with additional benefits of reduced computation and memory requirements at inference. DOC-RAG shows the best performance by combining both the contributions to adaptively weigh augmented predictions with LM output. We also observe that an increase in hyperparameter λ corresponds to an increase in perplexity scores as explicit memorization of rare word patterns extracted from similar domains benefits the NWP task. However, it steadily decreases after reaching an inflection point.

Adaptation to Unseen Domains: We observe that retrieval augmentation on fine-tuned models shows an increase of 5-18% compared to non-fine-tuned counterparts. This observation supports our hypothesis that transfer learning improves model performance on the out-of-domain test sets. Moreover, we see that explicit memorization from the out-of-domain train set is pivotal to effectively predict domain-specific rare word patterns missed during supervised fine-tuning step.

Runtime and Memory Cost: Let us assume the time complexity for a single pass through LM without augmentation is constant $O(C)$. Let us assume N domain-specific documents for any input sample. The vocab size of the dataset is V. Each document Neural Retriever model computes the relevance score for N documents and the query with overall time complexity of $O(NC)$. Bi-gram matrix computation for N documents can be approximated to $O(NV)$ considering each document may contain at most some multiple of V tokens. However, these bi-gram matrices are cached and their computation needs to be done only once for the entire external data. Finally, computing the augmented probability scores requires $O(NxV)$ time complexity, where V is the vocab size. Hence, overall time complexity is $O(NC + N + NV)$. Therefore, time Complexity of

	Model	WikiText-103 Perplexity (↓)	Earnings-21+22 Perplexity (↓)
w \o Fine-tuning	LSTM	1384.1	757.6
	+ Neural Cache	1325.3	723.8
	+ kNN-LM	1191.6	659.1
	+ RAG	585.4	452.5
	+ DOC-RAG	**539.8**	**412.3**
	+ DOC-RAG w \o Distributed Co-occurrence	603.6	477.2
	+ DOC-RAG w \o Contrastive Retriever	544.3	420.3
With Fine-tuning	LSTM	103.9	66.2
	+ Neural Cache	97.6	66.0
	+ kNN-LM	91.8	65.7
	+ RAG	85.3	63.1
	+ DOC-RAG	**80.2***	**59.6***
	+ DOC-RAG w \o Distributed Co-occurrence	89.2	64.5
	+ DOC-RAG w \o Contrastive Retriever	84.2	63.0

(a) LSTM LM

	Model	WikiText-103 Perplexity (↓)	Earnings-21+22 Perplexity (↓)
w \o Fine-tuning	Transformer	1322.3	834.2
	+ Neural Cache	1295.3	802.4
	+ kNN-LM	1150.4	717.8
	+ RAG	555.1	463.5
	+ DOC-RAG	**569.3***	**446.1***
	+ DOC-RAG w \o Distributed Co-occurrence	585.3	454.8
	+ DOC-RAG w o Contrastive Retriever	578.2	452.7
With Fine-tuning	Transformer	88.6	55.2
	+ Neural Cache	86.8	54.9
	+ kNN-LM	79.3	54.2
	+ RAG	75.3	52.5
	+ DOC-RAG	**72.5***	**49.6***
	+ DOC-RAG w \o Distributed Co-occurrence	76.5	53.8
	+ DOC-RAG w \o Contrastive Retriever	76.1	52.6

(b) Transformer LM

	Model	AMI Meeting Corpus Perplexity (↓)	AMI Meeting Corpus WER (↓)	TED LIUMv3 Perplexity (↓)	TED LIUMv3 WER (↓)
w \o Fine-tuning	Audio Model Only (Emformer)	–	32.54	–	17.23
	Audio Model + LSTM	1636.4	31.75	427.7	13.51
	+ Neural Cache	1545.4	31.69	414.5	13.25
	+ kNN-LM	1232.2	31.62	389.7	7.82
	+ RAG	535.1	31.43	336.6	7.35
	+ DOC-RAG	**471.2***	**31.15***	**315.0***	**7.16***
	+ DOC-RAG w \o Distributed Co-occurrence	606.7	31.25	335.4	7.34
	+ DOC-RAG w \o Contrastive Retriever	490.5	31.22	332.8	7.23
With Fine-tuning	Audio Model Only (Emformer)	–	32.54	–	17.23
	Audio Model + LSTM	37.7	31.40	132.6	13.27
	+ Neural Cache	37.5	31.36	132.2	13.03
	+ kNN-LM	37.1	31.27	131.5	7.76
	+ RAG	36.5	31.20	131.0	7.36
	+ DOC-RAG	**35.1***	**31.14***	**128.7***	**7.03***
	+ DOC-RAG w \o Distributed Co-occurrence	36.6	31.20	130.3	7.44
	+ DOC-RAG w \o Contrastive Retriever	36.2	31.16	130.2	7.28

(c) LSTM LM

	Model	AMI Meeting Corpus Perplexity (↓)	AMI Meeting Corpus WER (↓)	TED LIUMv3 Perplexity (↓)	TED LIUMv3 WER (↓)
w \o Fine-tuning	Audio Model Only (Emformer)	–	32.54	–	17.23
	Audio Model + Transformer	2114.3	32.05	442.0	13.24
	+ Neural Cache	1987.5	32.01	424.5	13.18
	+ kNN-LM	1579.0	31.95	398.6	7.57
	+ RAG	865.2	31.39	330.6	7.20
	+ DOC-RAG	**601.4***	**31.25***	**310.1***	**7.05***
	+ DOC-RAG w \o Distributed Co-occurrence	637.1	31.37	332.3	7.22
	+ DOC-RAG w \o Contrastive Retriever	624.9	31.33	327.4	7.14
With Fine-tuning	Audio Model Only (Emformer)	–	32.54	–	17.23
	Audio Model + Transformer	29.5	31.28	116.7	12.98
	+ Neural Cache	29.3	31.24	116.2	12.78
	+ kNN-LM	29.1	31.19	115.6	7.35
	+ RAG	27.4	31.09	114.2	7.15
	+ DOC-RAG	**26.4***	**31.03***	**112.3***	**6.93***
	+ DOC-RAG w \o Distributed Co-occurrence	28.1	31.14	114.0	7.21
	+ DOC-RAG w \o Contrastive Retriever	27.7	31.10	113.0	7.04

(d) Transformer LM

Figure 2: Performance comparison of DOC-RAG for (a,c) LSTM and (b,d) Transformer LMs and ablations (in red) for the **Next Word Prediction** and **Second-Pass ASR Re-scoring** tasks on (1) WikiText-103, (2) Earnings-21+22, (3) AMI Meeting Corpus, (4)TED LIUMv3 datasets. DOC-RAG achieves the lowest perplexity scores and minimum WER in all settings. * indicates statistically significant results based on Wilcoxon's signed rank test (5 runs, $p < 0.001$).

DOC-RAG at inference: $O(N(C+V))$; time complexity for Bi-gram matrix computation: $O(NV)$; memory for DOC-RAG cache: $O(NV^2)$.

Time and memory complexity for RAG with DPR is similar to DOC-RAG as it still needs to compute the relevance score without training the neural retriever from scratch $(O(constant + NV)$ which approximates to $O(NV))$. In Knn-LM, the datastore caches all context vectors for the entire train set. Each context vector requires a single pass through the BERT encoder, taking an over time complexity of at most $O(NV * C)$. Each context vector is of fixed dimension D (D=768 for BERT). So we compute context vectors for all tokens in the training set $(O(NV))$. Therefore, time complexity of Knn-LM at inference: $O(NV)$; time complexity of data store computation: $O(NVC)$; memory for Knn-LM: $O(NV * D)$.

DOC-RAG is more time efficient both during data store computation as it does not require a pass through encoder for each token in the data set. DOC-RAG has a slightly more time complexity due to domain ranking. Although it may seem that DOC-RAG requires more memory than KNN-LM, a large majority of the bi-gram matrices are sparse due to their cells being close to zero. Hence, we use Numpy sparse matrix implementation to compress their memory footprint. This is not possible in KNN-LM due to the high dimensionality of BERT embeddings that cannot be further compressed without loss of information.

5. Conclusion and Future Work

We introduce Domain-Distributed Co-occurrence Retrieval Augmentation (DOC-RAG) for ASR LM personalization. This technique involves a contrastively trained retrieval module ranking external knowledge domains based on their semantic similarity with the input query. We use bi-gram word frequency distribution to recognize personalized word patterns associated with specific users/domains and aggregate the contextual probabilities of the next word prediction task from different domains through relative augmentation of the input query. Experiments on four user-specific ASR corpora show that DOC-RAG achieves the best perplexity and WER. our proposed method is easily extensible to any encoder, including large Transformer decoder models like LLama (Touvron et al., 2023). by just using the next word prediction probabilities from such large models. This profound advantage of our method helps make our work relevant even for future ASR LLM decoder models where we can utilize DOC-RAG augmentation without any architectural changes. Our method can also be directly utilized for improving LLM text generation performance for novel unseen domains. Future work will explore multilingual and streaming ASR.

6. References

Jesús Andrés-Ferrer, Dario Albesano, Puming Zhan, and Paul Vozila. 2022. Contextual density ratio for language model biasing of sequence to sequence asr systems. *Interspeech*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Theresa Breiner, Swaroop Ramaswamy, Ehsan Variani, Shefali Garg, Rajiv Mathews, Khe Chai Sim, Kilol Gupta, Mingqing Chen, and Lara McConnaughey. 2022. Userlibri: A dataset for asr personalization using only text. *Interspeech*.

Ciprian Chelba, Mohammad Norouzi, and Samy Bengio. 2017. N-gram language modeling using recurrent neural network estimation. *arXiv preprint arXiv:1703.10724*.

Xie Chen, Xunying Liu, Mark JF Gales, and Philip C Woodland. 2015. Recurrent neural network language model training with noise contrastive estimation for speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5411–5415. IEEE.

Shih-Hsuan Chiu, Tien-Hang Lo, and Berlin Chen. 2021. Cross-sentence neural language models for conversational speech recognition. In *2021 International Joint Conference on Neural Networks (IJCNN)*.

Nilaksh Das, Duen Horng Chau, Monica Sunkara, Sravan Bodapati, Dhanush Bekal, and Katrin Kirchhoff. 2022. Listen, know and spell: Knowledge-infused subword modeling for improving asr performance of oov named entities. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Steven B. Davis and Paul Mermelstein. 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357–366.

Miguel Del Rio, Peter Ha, Quinten McNamara, Corey Miller, and Shipra Chandra. 2022. Earnings-22: A practical benchmark for accents in the wild. *arXiv preprint arXiv:2203.15591*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.

Edouard Grave, Moustapha Cissé, and Armand Joulin. 2017. Unbounded cache model for online language modeling with open vocabulary. *ArXiv*, abs/1711.02604.

Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning – Data Mining, Inference, and Prediction*. Springer, New York.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Esteve. 2018. 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.

Kazuki Irie, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019a. Language modeling with deep transformers. In *Interspeech*.

Kazuki Irie, Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2019b. Training language models for long-span cross-sentence evaluation. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 419–426.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. 2022. Towards continual knowledge learning of language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *ACM Computing Surveys*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *Annual Meeting of the Association for Computational Linguistics*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Diederik P. Kingma and Jimmy Ba. 2015. 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*.

Wessel Kraaij, Thomas Hain, Mike Lincoln, and Wilfried Post. 2005. The ami meeting corpus.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Ke Li, Zhe Liu, Tianxing He, Hongzhao Huang, Fuchun Peng, Daniel Povey, and Sanjeev Khudanpur. 2020a. An empirical study of transformer-based neural language model adaptation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7934–7938. IEEE.

Ke Li, Daniel Povey, and Sanjeev Khudanpur. 2020b. Neural language modeling with implicit cache pointers. In *Interspeech*.

Qi Liu, Dani Yogatama, and Phil Blunsom. 2022. Relational memory-augmented language models. *Transactions of the Association for Computational Linguistics*, 10:555–572.

Richard Diehl Martinez, Scott Novotney, Ivan Bulyko, Ariya Rastrow, Andreas Stolcke, and Ankur Gandhe. 2021. Attention-based contextual language model adaptation for speech recognition. *ACL Findings*.

Puneet Mathur, Zhe Liu, Ke Li, Yingyi Ma, Gil Keren, Zeeshan Ahmed, Dinesh Manocha, and Xuedong Zhang. 2023. Personalm: Language model personalization via domain-distributed span aggregated k-nearest n-gram retrieval augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11314–11328.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari.

Roger K. Moore and Lucy Skidmore. 2019. On the use/misuse of the term 'Phoneme'. In *Proc. INTERSPEECH 2019 – 20th Annual Conference of the International Speech Communication Association*, pages 2340–2344, Graz, Austria.

Aakanksha Naik, Jill Lehman, and Carolyn Rosé. 2022. Adapting to the long tail: A meta-analysis of transfer learning research for language understanding tasks. *Transactions of the Association for Computational Linguistics*, 10:956–980.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language*

Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 2463–2473. Association for Computational Linguistics.

Lawrence R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Miguel Del Rio, Natalie Delworth, Ryan Westerman, Michelle Huang, Nishchal Bhandari, Joseph Palakapilly, Quinten McNamara, Joshua Dong, Piotr Zelasko, and Miguel Jette. 2021. Earnings-21: A practical benchmark for asr in the wild.

Timo Schick and Hinrich Schütze. 2019. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *AAAI Conference on Artificial Intelligence*.

Prashant Serai, Vishal Sunder, and Eric Fosler-Lussier. 2022. Hallucination of speech recognition errors with sequence to sequence learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:890–900.

Yangyang Shi, Yongqiang Wang, Chunyang Wu, Ching-Feng Yeh, Julian Chan, Frank Zhang, Duc Le, and Mike Seltzer. 2021. Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*. IEEE.

Adam Stooke, Khe Chai Sim, Mason Chua, Tsendsuren Munkhdalai, and Trevor Strohman. 2023. Internal language model personalization of e2e automatic speech recognition using random encoder features. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 213–220. IEEE.

G. Sun, C. Zhang, and P. C. Woodland. 2021. Transformer language models with lstm-based cross-utterance information representation. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7363–7367.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Emiru Tsunoo, Yosuke Kashiwagi, Chaitanya Narisetty, and Shinji Watanabe. 2022. Residual language model for end-to-end speech recognition. *Interspeech*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zecheng Wang and Yik-Cheung Tam. 2022. Suffix retrieval-augmented language modeling. *ArXiv*, abs/2211.03053.

Hainan Xu, Ke Li, Yiming Wang, Jian Wang, Shiyin Kang, Xie Chen, Daniel Povey, and Sanjeev Khudanpur. 2018. Neural network language modeling with letter-based features and importance sampling. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6109–6113. IEEE.