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 re-train 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
Domain-distributed Co-occurrence Retrieval Augmentation: (Left) Input query $q$ and domain-specific text/recordings are encoded using BERT models $(\eta_1, \eta_2)$, 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 $f_i(w_l,w_{l+1})$. We sum the word co-occurrence probabilities weighted by the relevance of the selected domain to obtain the probability $P(w_l|q)$ for next-word prediction.

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 $\eta_1$ and $\eta_2$, 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 embeddings $c_j$ 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 $\tau$ 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

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 $C$) based on the relevance of $K$ unseen domains/users $d_1, d_2, \ldots, d_K$ to the input query $q = \{w_1, w_2, \ldots, w_{L-1}\}$. 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 $P_{LM}(w_l|q)$ over the target token $w_l$ 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 $P_{DOC-RAG}(w_l|q) = \sum_{i=1}^{K} P(d_i|q) \times P(w_l|d_i, q)$. 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 rescoring based on language model perplexity (ASR 2nd-pass rescoring).
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 $f_{i}^{j}(w_{t-1}, w_{t})$ 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 $P(w_{t}|d_i, q) = \frac{\sum_{i=0}^{R} s(q, d_i) \cdot [f_{i}^{j}(w_{t-1,1}) \cdot f_{i}^{j}(w_{t-1,2}) \cdots f_{i}^{j}(w_{t-1,V})]}{\sum_{i=0}^{R} s(q, d_i) \cdot \sum_{i=0}^{R} s(q, d_i) \cdot f_{i}^{j}(w_{t-1,1}) \cdots f_{i}^{j}(w_{t-1,V})}$.

**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 $P_{DOC-RAG}(w_{t}|q) = \sum_{i=0}^{R} s(q, d_i) \cdot [f_{i}^{j}(w_{t-1,1}) \cdot f_{i}^{j}(w_{t-1,2}) \cdots f_{i}^{j}(w_{t-1,V})]$.

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 $\lambda$ 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) (Panayotov et al., 2015), and two speech datasets - AMI Meetings (Kraaij et al., 2005), speaker TED talks (Hernandez et al., 2022); Del Rio et al., 2022). To study the personalization of ASR LMs, we re-formulated 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.

**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.

**Ablation Studies:** We run the following ablation studies:

- **(i) DOC-RAG:** We evaluate the following ablation studies:
  - **(ii) 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.
Table 1: Data Statistics

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

- (ii) **DOC-RAG**: 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 $\lambda$ 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 deteriorates 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 $\lambda$ 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
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: $O(NV^2 + D)$.

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


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


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*.


