

QUERY DRIFT COMPENSATION: ENABLING COMPATIBILITY IN CONTINUAL LEARNING OF RETRIEVAL EMBEDDING MODELS

Dipam Goswami^{1,2} Liying Wang^{1,2} Bartłomiej Twardowski^{1,2,3} Joost van de Weijer^{1,2}

¹ Computer Vision Center, Barcelona, Spain

² Department of Computer Science, Universitat Autònoma de Barcelona, Spain

³ IDEAS Research Center, Warsaw, Poland

{dgoswami, liying, btwardowski, joost}@cvc.uab.es

ABSTRACT

Text embedding models enable semantic search, powering several NLP applications like Retrieval Augmented Generation by efficient information retrieval (IR). However, text embedding models are commonly studied in scenarios where the training data is static, thus limiting its applications to dynamic scenarios where new training data emerges over time. IR methods generally encode a huge corpus of documents to low-dimensional embeddings and store them in a database index. During retrieval, a semantic search over the corpus is performed and the document whose embedding is most similar to the query embedding is returned. When updating an embedding model with new training data, using the already indexed corpus is suboptimal due to the non-compatibility issue, since the model which was used to obtain the embeddings of the corpus has changed. While re-indexing of old corpus documents using the updated model enables compatibility, it requires much higher computation and time. Thus, it is critical to study how the already indexed corpus can still be effectively used without the need of re-indexing. In this work, we establish a continual learning benchmark with large-scale datasets and continually train dense retrieval embedding models on query-document pairs from new datasets in each task and observe forgetting on old tasks due to significant drift of embeddings. We employ embedding distillation on both query and document embeddings to maintain stability and propose a novel query drift compensation method during retrieval to project new model query embeddings to the old embedding space. This enables compatibility with previously indexed corpus embeddings extracted using the old model and thus reduces the forgetting. We show that the proposed method significantly improves performance without any re-indexing. Code is available at <https://github.com/dipamgoswami/QDC>.

1 INTRODUCTION

Information Retrieval (IR) is widely used in several NLP applications like semantic search and Retrieval-Augmented Generation (RAG) for LLMs. Text embeddings which encode a sentence or a chunk of text to low-dimensional embedding vectors are commonly used for these applications (Lewis et al., 2020; Ram et al., 2023; Izacard et al., 2023). Semantic search enables us to retrieve most relevant responses from a document corpus for a given query. While non-semantic lexical approaches like TF-IDF and BM25 (Robertson et al., 2009) were traditionally used for retrieval, dense retrievers like transformer architectures (Vaswani et al., 2017) are now widely used for semantic search (Cer et al., 2018; Yates et al., 2021; Thakur et al., 2021; Muennighoff, 2022; Nussbaum et al., 2024). While lexical methods consider queries and documents as bag-of-words, dense retriever models encode the queries and corpus documents in a shared semantic embedding space (Gillick et al., 2018) which enables us to precompute and index the document embeddings from the corpus before performing retrieval. The standard practice (Thakur et al., 2021; Muennighoff, 2022; Nussbaum et al., 2024) is to consider a static setting in which the retriever model is trained on several datasets and the corpus documents are indexed using the trained retriever model. These indexed corpus document embeddings are then used for evaluation of retrieval for each task or dataset separately. However, in many practical applications not all datasets are jointly available, and new data arrives over time. In order to improve the retrieval models on newly incoming datasets, one needs to continually train them. In this work, we study how the retriever models can be fine-tuned on new datasets over time and how updating the model can impact the retrieval process and performance.

Continual Learning (CL) enables neural networks to learn a sequence of tasks one after another and perform well on all seen tasks. Several research works (De Lange et al., 2021; Masana et al., 2022; Bornschein et al., 2023; Wang et al., 2024; Zhou et al., 2024; Verwimp et al., 2024) explore various aspects of CL, primarily for image classification

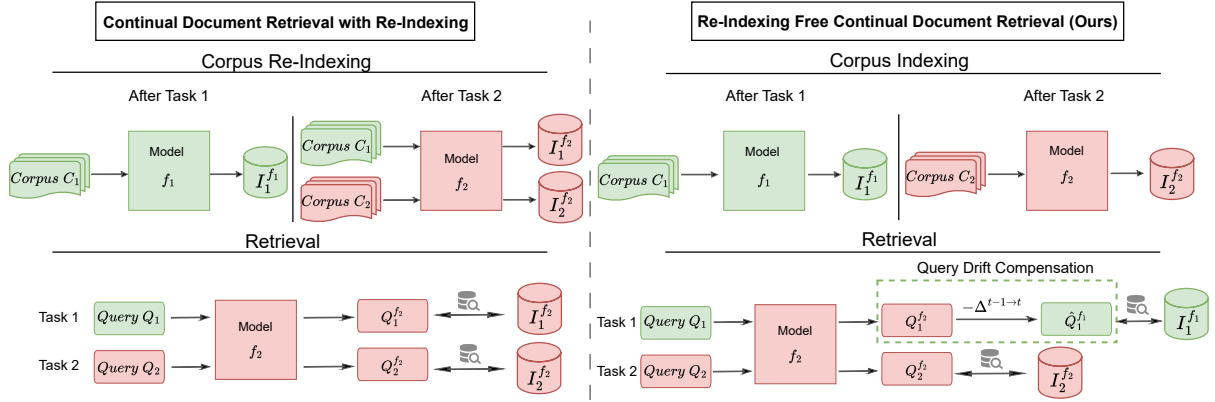


Figure 1: Continual Document Retrieval (CDR). Here, we consider a two-task continual setting and illustrate two different approaches to tackle the *embedding drift* issue for old task retrieval in CDR which arises due to non-compatibility between query embeddings from the updated model $Q_1^{f_2}$ (in red) and corpus embeddings indexed using the old model $I_1^{f_1}$ (in green). (left) A naive approach to make the query and corpus embeddings compatible is by re-indexing the corpus documents using the updated model. However, re-indexing large amounts of documents from all old tasks after updating the model on every new task is time-consuming and computationally expensive. (right) To avoid re-indexing, we propose to estimate embedding drift from old to new model $\Delta^{t-1 \rightarrow t}$ and compensate the drift from $Q_1^{f_2}$ during retrieval. The proposed query drift compensation approach enables compatibility by projecting the query embedding back to the embedding space of f_1 without any need for expensive re-indexing.

tasks. A major challenge in CL is catastrophic forgetting (McCloskey & Cohen, 1989; Kemker et al., 2018) which refers to forgetting old tasks after learning new tasks. We focus on the task-incremental learning setting (Van de Ven & Tolias, 2019), where the task-id information is available during inference or retrieval. In this work, we discuss how continually updating the retriever models for document retrieval could lead to issues of non-compatibility between query and corpus document embeddings for old tasks. This is due to the fact that during retrieval of queries from old tasks, the query embeddings are obtained using the updated model while the corpus embeddings were previously indexed using the old model from the respective tasks. The issue of non-compatibility has previously been studied in CL for image retrieval works (Ramanujan et al., 2022; Wan et al., 2022; Biondi et al., 2023). The drop in performance of old tasks when naively fine-tuning on new tasks can be attributed to the *embedding drift* (also referred to as semantic drift (Yu et al., 2020)) between old and new embedding spaces, as we demonstrate in our experiments.

One naive approach could be re-indexing of corpus documents from all old tasks (also known as back-filling in image retrieval (Shen et al., 2020)) to ensure that the query and the corpus document embeddings are in the same embedding space as shown in Fig. 1 (left). However, re-indexing of all old corpus documents after each task would involve very high computation costs. In order to avoid the re-indexing of old documents, we propose a novel query drift compensation approach which projects the query embeddings from the new model back to the embedding space of their respective tasks as illustrated in Fig. 1 (right). We estimate query drift vectors for each task transition using the query samples from training data of the current task which we use to approximate the query drift of previous tasks. We compensate or subtract these drift vectors from new model query embeddings to move them back to the old space. Thus, instead of re-indexing old documents, we propose a simple query projection which makes the query and document embeddings compatible and thus reduces the forgetting of old tasks significantly.

We propose a CL benchmark for document retrieval tasks where we train on new datasets in each task using query-document pairs (see Table 1) and aim to improve the document retrieval performance on all old and new tasks. We fine-tune the retriever model on several datasets from the BEIR benchmark (Thakur et al., 2021) to improve retrieval on specific data or domains. We also propose to employ embedding distillation between new and the previous task model to reduce the *embedding drift* and the forgetting of old tasks. Our contributions can be summarized as:

- We establish a benchmark for Continual Document Retrieval (CDR) to evaluate the effect of continual training on large-scale datasets and observe forgetting of old tasks after fine-tuning on new tasks. We show that knowledge distillation using both query and corpus embeddings during fine-tuning can reduce the forgetting.
- We discuss how non-compatibility due to the *embedding drift* hurts retrieval performance. We propose a novel approach - Query Drift Compensation (QDC) which estimates the embedding drift between tasks after training on a new task. During retrieval, we propose to compensate the drift from queries (extracted from the updated model) to enable compatibility with the document embeddings (indexed using the old model).

Table 1: Examples of query-document pairs of different datasets from the BEIR (Thakur et al., 2021) benchmark.

| Dataset | Query | Document |
|-------------------------------|--|---|
| MS MARCO (Bajaj et al., 2016) | how much magnesium in kidney beans | Kidney Beans. A cup of kidney beans contains 70 mg of magnesium and is a great source of protein and fiber. More: How to Bake With Beans. |
| NQ (Kwiatkowski et al., 2019) | what is the meaning of a crown | Crowns are the main symbols of royal authority.[21] |
| HotpotQA (Yang et al., 2018) | What compound, known as aqua fortis or spirit of niter is used in rocket propellant? | Nitric acid (HNO ₃), also known as aqua fortis and spirit of niter, is a highly corrosive mineral acid. |
| FEVER (Thorne et al., 2018) | what team won the az state peach bowl | The 1970 Peach Bowl was a college football bowl game between the Arizona State Sun Devils and the North Carolina Tar Heels . |
| FiQA-2018 (Maia et al., 2018) | How can a 'saver' maintain or increase wealth in low interest rate economy? | Personally, I invest in mutual funds. Quite a bit in index funds, some in capital growth & international. |

- We demonstrate in our experiments how the proposed QDC enables continual learning of retrieval models on new datasets without any re-indexing of old documents and performs similar to joint training. We also show that our approach outperforms re-indexing based CDR.

2 RELATED WORKS

Continual Learning. CL methods (De Lange et al., 2021; Masana et al., 2022; Wang et al., 2024; Zhou et al., 2024) focus primarily on reducing catastrophic forgetting of old classes after learning new classes. CL methods can be classified into class-incremental, task-incremental and domain-incremental settings (Van de Ven & Tolias, 2019). In this work, we focus on the task-incremental setting where we have access to task-id during inference, since it is a more realistic setting for retrieval. Existing CL approaches can be broadly divided into replay-based, regularization-based, parameter isolation-based and prototype-based methods. Replay-based methods (Rebuffi et al., 2017; Belouadah & Popescu, 2019) store exemplars from old tasks and use them during training on new tasks. However, storing exemplars has several limitations due to data regulations and privacy issues, as discussed in Goswami et al. (2024). Regularization-based methods prevent updates of important weights for old classes (Kirkpatrick et al., 2017) or employ knowledge distillation approaches (Li & Hoiem, 2018; Douillard et al., 2020) between previous and current model to preserve knowledge from previous tasks. Some methods (Mallya & Lazechnik, 2018; Serra et al., 2018) divide the model to learn task-specific parameters. Prototype-based methods (Yu et al., 2020; Goswami et al., 2023) store a prototype representation for all seen classes and classify samples based on distances to the class prototypes. Semantic drift compensation (Yu et al., 2020; Goswami et al., 2024; Gomez-Villa et al., 2025) has been used to improve prototype-based methods by updating old prototypes. We use the concept of drift compensation for backward projection of queries from latest model to old model embedding space, thus enabling query-corpus compatibility.

Document Retrieval. Following success of large language models, the use of neural retrieval models (Karpukhin et al., 2020; Liang et al., 2020; Khattab & Zaharia, 2020; Luan et al., 2021; Muennighoff, 2022) has become more common than traditional lexical approaches (Robertson et al., 2009), which have the lexical gap (Berger et al., 2000). Dense retrieval models (Gillick et al., 2018) map queries and documents in a shared dense embedding space, and scores their relevance based on cosine similarity between query and document embeddings. Recent works (Günther et al., 2023; Nussbaum et al., 2024) use modified BERT (Devlin et al., 2019) and perform MLM pretraining followed by unsupervised contrastive finetuning on large corpus of data and finally finetune on labelled datasets. In this work, we use the nomic embedding model from Nussbaum et al. (2024) which outperforms several competitive models (Izacard et al., 2021; Wang et al., 2022; Li et al., 2023; Günther et al., 2023) in retrieval tasks.

Continual Document Retrieval. In CDR, the retriever model is expected to continually learn from new tasks over time without forgetting the previous tasks. While several works studied continual image retrieval (Wan et al., 2022; Biondi et al., 2024) and continual multimodal retrieval (Wang et al., 2021), fewer studies have explored continual learning in information retrieval. Lovón-Melgarejo et al. (2021) investigated the catastrophic forgetting problem in neural ranking models for information retrieval. Gerald & Soulier (2022) built a continual information retrieval setting using a single dataset MS MARCO (Bajaj et al., 2016) and observed that catastrophic forgetting exists in IR to a lesser extent than image classification tasks. Cai et al. (2023) proposed a re-indexing free memory-based method for first-stage retrieval in a different setting with unlabeled new task documents. Recently, Hou et al. (2025) investigated how existing CL methods work in CDR by dividing MS MARCO into multiple tasks based on topics. Another set of works (Kishore et al., 2023; Mehta et al., 2023) explored differentiable search index for CDR on how to encode the corpus of documents in the model parameters where the model output for a given query is the predicted document. In this work, we propose a more comprehensive benchmark for CDR using five large-scale datasets from Thakur et al. (2021) and analyze how continual training affect the retrieval performance.

3 CONTINUAL LEARNING FOR DOCUMENT RETRIEVAL

Here, we introduce and formalize the setting of Continual Document Retrieval (CDR). Following [Thakur et al. \(2021\)](#), we refer to a text of any length from the corpus as a ‘document’. Document retrieval aims to return the most relevant document d from the given corpus C as a response to the query q provided by the user. In the continual setting, we refer to the set of queries and corpus of documents for task $t \in [1, T]$ as Q_t and C_t . Here, we denote a single query and document from task t as q_t and d_t respectively, where $d_t \in C_t$. In the first task, we train the embedding model f_1 on query-document pairs $\{Q_1, D_1\}$ from the training set of task 1. Note that $D_1 \in C_1$ refers to the set of documents from the corpus of task 1 that are used for training (typically the entire corpus is larger). We use the trained model f_1 for indexing all the documents from C_1 and refer to the indexed document embeddings as $I_1^{f_1}$. Similarly, we also use $q_j^{f_i} = f_i(q_j)$. Here, the superscript term denotes the model used for indexing and the subscript term denotes the task to which the document belongs. Similarly, for task t , we train embedding model f_t on $\{Q_t, D_t\}$ pairs from the training set of task t and the corpus documents indexed by f_t are referred to as $I_t^{f_t}$. In our setting, we consider that during the training of task t we have no access to any data from previous tasks (also known as exemplar-free continual learning ([Smith et al., 2023](#); [Petit et al., 2023](#); [Goswami et al., 2023](#); [Magistri et al., 2024](#))).

Non-compatibility due to Embedding Drift. After task t , we have access to the updated model f_t only, and thus we need to use f_t to embed the queries from all tasks during retrieval phase. For an old task $t' < t$, the queries $Q_{t'}$ from task t' are embedded using f_t denoted as $Q_{t'}^{f_t}$ but all the corpus documents $C_{t'}$ from t' were already indexed after task t' using $f_{t'}$ and stored as $I_{t'}^{f_{t'}}$. We refer to this phenomenon as *embedding drift*. This leads to non-compatibility between query and document embeddings, since they are in two different embedding spaces (the embedding space has been updated during the continual training). We illustrate the setting of continual document retrieval and the non-compatibility issue in Fig. 1.

Re-indexing of old task corpus. A naive approach to enable compatibility of query and document embeddings is to re-index all the corpus documents from all old tasks to obtain $I_{t'}^{f_t} \forall t' < t$ after training on a new task t . Re-indexing removes the embedding drift between the queries $Q_{t'}^{f_t}$ and corpus documents $I_{t'}^{f_t}$, which are then in the same embedding space thus resolving the non-compatibility issue (see Fig. 1).

However, re-indexing involves high computation costs and it potentially takes a considerable amount of time to re-index large amounts of documents from all old tasks after finetuning on every new task. Therefore, in this paper, we focus on re-indexing free CDR. Re-indexing (or back-filling) based approach has been found to be effective and considered an upper bound in fine-tuning of image retrieval models ([Shen et al., 2020](#)). Interestingly, from our analysis in CDR, we show that re-indexing based CDR performs poorly. We attribute this to misalignment due to the unequal drift of query and corpus embeddings (similar to observations in multi-modal continual retrieval ([Wang et al., 2021](#))). We revisit and analyze this in the experiments section.

4 RE-INDEXING FREE CONTINUAL DOCUMENT RETRIEVAL

4.1 TRAINING STRATEGY

We follow the training strategy from [Nussbaum et al. \(2024\)](#) which performs masked language modeling (MLM) pre-training to train a long-context BERT model followed by multi-stage contrastive learning ([Li et al., 2023](#)). The first stage is unsupervised contrastive pre-training using the InfoNCE contrastive loss ([Oord et al., 2018](#)) on huge amounts of publicly available text-pairs which are mined from the web. Following the pre-training stages, the final step is performing supervised contrastive finetuning on each dataset at each task. Here, we consider the model pre-trained using MLM and unsupervised contrastive learning and then continually finetune the pre-trained model with supervised contrastive learning on query-document (q, d) training pairs of each task.

For the continual training at task t , we initialize a new model f_t with the weights of f_{t-1} and then perform contrastive training on (q, d) pairs from the training set of task t . For each (q, d) pair, we consider the most similar documents as hard negatives and use \mathcal{H} hard negatives for each query which are mined from the same dataset. We minimize the contrastive loss function for a given batch of (q, d) pairs as follows:

$$\mathcal{L}_C = -\frac{1}{n} \sum_i \log \frac{e^{S(f_t(q_i), f_t(d_i))/\tau}}{\sum_{j=1}^n e^{S(f_t(q_i), f_t(d_j))/\tau} + \sum_{h=1}^{\mathcal{H}} e^{S(f_t(q_i), f_t(d_h))/\tau}}, \quad (1)$$

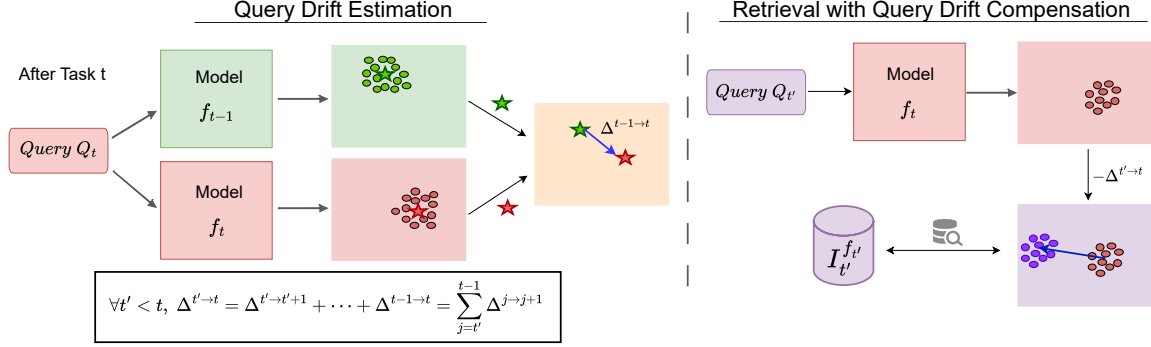


Figure 2: Illustration of Query Drift Compensation. (left) We show how to estimate the query drift vectors $\Delta^{t-1 \rightarrow t}$ for each task transition (from $t-1$ to t) after training on task t . We store the drift vectors of all old tasks. For an old task t' ($t' < t$), we obtain the drift vector $\Delta^{t' \rightarrow t}$ by addition of all drift vectors from task t' to t . (right) During retrieval of old task t' , we compensate the query embeddings with drift vector $\Delta^{t' \rightarrow t}$ to project them from embedding space of task t to that of t' . As a result, we compare the query and previously indexed document embeddings in the same embedding space of task t' (in purple), thus avoiding the non-compatibility issue.

where $S(q, d)$ is the cosine similarity, τ is the temperature parameter and n is the batch size. Here, the first term in the denominator $\sum_{j=1}^n e^{S(f_t(q_i), f_t(d_j)) / \tau}$ includes the in-batch negatives when $j \neq i$ and the second term $\sum_{h=1}^{\mathcal{H}} e^{S(f_t(q_i), f_t(d_h)) / \tau}$ includes the hard-negatives for the corresponding query.

4.2 EMBEDDING KNOWLEDGE DISTILLATION

A naive application of Eq. (1) for all the tasks would lead to catastrophic forgetting and an embedding which is especially tailored to the last task. Therefore, here we adapt a commonly used regularization technique in CL for CDR to prevent forgetting (Li & Hoiem, 2018). More particularly, to reduce the feature drift and improve stability, we propose to perform feature distillation (Yu et al., 2020) by minimizing the cosine distance between the embeddings from old and new models. We perform the distillation on embeddings of both queries and documents from training data of the new task. We minimize the following distillation loss for a given batch of (q, d) pairs:

$$\mathcal{L}_D = \frac{1}{n} \sum_{i=1}^n (D_c(f_t(q_i), f_{t-1}(q_i)) + D_c(f_t(d_i), f_{t-1}(d_i))), \quad (2)$$

where D_c is the cosine distance between embeddings and n is the batch size. Even though the distillation improves the stability of the network, it does not completely solve the non-compatibility issue discussed before, since the learning of new tasks still requires the embedding space to adapt and incorporate new knowledge.

4.3 QUERY DRIFT COMPENSATION

Here we propose Query Drift Compensation (QDC) which addresses the non-compatibility issue in re-indexing free CDR. In this case, for $t' < t$ we would like to query the corpus $I_{t'}^{f_{t'}}$ with the query embedding from the same embedding model $Q_{t'}^{f_{t'}}$, however we have access to the query embedding using the latest embedding model $Q_{t'}^{f_t}$, which leads to the non-compatibility issue. To enable backward compatibility of old task query embeddings $Q_{t'}^{f_t}$, $\forall t' < t$ (indexed using continually updated new model f_t) with the corpus embeddings indexed using the old model $f_{t'}$, we propose to use drift compensation inspired by SDC (Yu et al., 2020) which was proposed for image classification in continual learning.

We define the difference between the query embeddings $q_{t'}^{f_{t'}}$ and $q_{t'}^{f_t}$ as the *query drift*:

$$\delta^{t' \rightarrow t} = q_{t'}^{f_t} - q_{t'}^{f_{t'}}, \quad (3)$$

If we have this query drift, the desired embedding $q_{t'}^{f_{t'}}$ can be easily computed with $q_{t'}^{f_{t'}} = q_{t'}^{f_t} - \delta^{t' \rightarrow t}$. However, we do not have access to $\delta^{t' \rightarrow t}$ as we cannot access the old task training data (the old task queries $q_{t'}$ cannot be used during task t). Therefore, in the following, we propose a method to approximate this drift based on the current task query drift.

Algorithm 1 Proposed Method for Continual Document Retrieval**Continual Training:****Input:** $t \in [1, T]$: task number; (Q_t, D_t) : training data; C_t : corpus f_0 : pre-trained model; \mathcal{H}, τ : hyper-parameters**Output:** f_t : Model trained in task t $\{I_z^{f_z}\} \forall z \in [1, t]$: Indexed corpus embeddings $\{\Delta^{z \rightarrow z+1}\} \forall z \in [1, t-1]$: drift vectors**for** task $t \in [1, T]$ **do** $f_t = f_{t-1}$ # f_{t-1} is frozen and f_t is trainable**for** each (q, d) pair in (Q_t, D_t) **do**

$$\mathcal{L}_C = -\frac{1}{n} \sum_i \log \frac{e^{S(f_t(q_i), f_t(d_i))/\tau}}{\sum_{j=1}^n e^{S(f_t(q_i), f_t(d_j))/\tau} + \sum_{h=1}^{\mathcal{H}} e^{S(f_t(q_i), f_t(d_h))/\tau}}$$

if $t > 1$ **then**

$$\mathcal{L}_D = \frac{1}{n} \sum_{i=1}^n D_c(f_t(q_i), f_{t-1}(q_i)) + D_c(f_t(d_i), f_{t-1}(d_i))$$

$$\mathcal{L} = \mathcal{L}_C + \mathcal{L}_D$$

else

$$\mathcal{L} = \mathcal{L}_C$$

end if**end for**# After f_t is trained, we estimate the drift vectors**if** $t > 1$ **then**

$$\Delta^{t-1 \rightarrow t} = \frac{1}{N_t} \sum_{q \in Q_t} (f_t(q) - f_{t-1}(q)) \quad \# \text{ Store } \Delta^{t-1 \rightarrow t}$$

end if# Indexing of corpus documents using trained model f_t

$$I_t^{f_t} \leftarrow f_t(d) \forall d \in C_t \quad \# \text{ Store document embeddings in } I_t^{f_t}$$

end for**Retrieval Evaluation after task t :****Input:** f_t : Model from task t $t' \in [1, t]$: task for evaluation $Q_{t'}$: test query data from task t' $C_{t'}$: corpus $\{I_z^{f_z}\} \forall z \in [1, t]$: Indexed corpus embeddings $\{\Delta^{z \rightarrow z+1}\} \forall z \in [1, t-1]$: drift vectors**Output:** $\hat{D}_{t'}$: retrieved documents corresponding to $Q_{t'}$ **for** task $t' \in [1, t]$ **do****if** $t' \leq t$ **then**

$$\Delta^{t' \rightarrow t} = \sum_{j=t'}^{t-1} \Delta^{j \rightarrow j+1}$$

$$\hat{Q}_{t'}^{f_{t'}} = f_t(q_{t'}) - \Delta^{t' \rightarrow t} \forall q_{t'} \in Q_{t'}$$

else

$$\hat{Q}_{t'}^{f_{t'}} = f_t(q_{t'}) \forall q_{t'} \in Q_{t'} \quad \# t' = t$$

end if**for** \hat{q} in $\hat{Q}_{t'}^{f_{t'}}$ **do**

$$\mathcal{I}_i = I_{t'}^{f_{t'}}[i]$$

$$index = \arg \max_i (S(\hat{q}, \mathcal{I}_i))$$

$$\hat{d} = C_{t'}[index]$$

$$\hat{D}_{t'} \leftarrow \hat{d} \quad \# \text{ Store retrieved documents in } \hat{D}_{t'}$$

end for**end for**

After learning a new task, we estimate the drift in the embedding space from f_{t-1} to f_t using the queries from the training data of the current task as shown in Fig. 2 (left). We project the queries $q \in Q_t$ through f_{t-1} and then through f_t . Now that we have the queries and their corresponding embeddings from f_{t-1} and f_t , we simply estimate the drift of queries and then average those drift vectors to obtain a single drift vector $\Delta^{t-1 \rightarrow t}$ for each task t as follows:

$$\Delta^{t-1 \rightarrow t} = \frac{1}{N_t} \sum_{q \in Q_t} (f_t(q) - f_{t-1}(q)), \quad (4)$$

where N_t is the number of queries in training data of task t . Thus, after learning a new task model f_t , we compute the query drift for task t and store the drift vector $\Delta^{t-1 \rightarrow t}$.

The drift vector from the last task can be simply added to drift vectors from previous tasks to obtain $\Delta^{t' \rightarrow t}$ as follows:

$$\Delta^{t' \rightarrow t} = \Delta^{t' \rightarrow t'+1} + \dots + \Delta^{t-1 \rightarrow t} = \sum_{j=t'}^{t-1} \Delta^{j \rightarrow j+1}, \quad (5)$$

During retrieval time, for task t' , we pass the queries $Q_{t'}$ through the updated model f_t to obtain embeddings $Q_{t'}^{f_t}$ and then compensate the embeddings by subtracting the drift vector of the corresponding task $\Delta^{t' \rightarrow t}$ to project them back to the embedding space of $f_{t'}$ as illustrated in Fig. 2 (right). For query $q_{t'} \in Q_{t'}$, we perform the query drift compensation as follows:

$$\hat{Q}_{t'}^{f_{t'}} = \hat{f}_{t'}(q_{t'}) = f_t(q_{t'}) - \Delta^{t' \rightarrow t}, \quad (6)$$

where $\hat{f}_{t'}(q_{t'})$ is the set of queries which are estimated using the proposed method and is expected to be similar to $f_{t'}(q_{t'})$. Having estimated $\hat{Q}_{t'}^{f_{t'}}$ for queries from task t' , we can now compare the query $\hat{q} \in \hat{Q}_{t'}^{f_{t'}}$ and indexed document embeddings $I_{t'}^{f_{t'}}$ in the same embedding space of $f_{t'}$ as follows:

$$\mathcal{I}_i = I_{t'}^{f_{t'}}[i]; \quad index = \arg \max_i (S(\hat{q}, \mathcal{I}_i)); \quad \hat{d} = C_{t'}[index], \quad (7)$$

where \mathcal{I}_i refers to embeddings from the indexed corpus $I_{t'}^{f_{t'}}$ at index i and S refers to the cosine similarity. Thus, we resolve the non-compatibility issue by simple vector subtraction without any re-indexing. We summarize the proposed approach for training and retrieval evaluation in Algorithm 1. In the experiments, we also explore estimating multiple query drift vectors per task, but do not find this to significantly improve results.

Table 2: Details of datasets used in the proposed benchmark for CDR. Details excerpted from Thakur et al. (2021).

| Split (\rightarrow) | | | Train | | Test | | Avg. Word Lengths | |
|-------------------------|-------------------------|--------------------------|---------|--------|-----------|------------|-------------------|----------|
| Task (\downarrow) | Domain (\downarrow) | Dataset (\downarrow) | #Pairs | #Query | #Corpus | Avg. D / Q | Query | Document |
| Passage-Retrieval | Misc. | MS MARCO | 532,761 | 6,980 | 8,841,823 | 1.1 | 5.96 | 55.98 |
| Question Answering (QA) | Wikipedia | NQ | 132,803 | 3,452 | 2,681,468 | 1.2 | 9.16 | 78.88 |
| | Wikipedia | HotpotQA | 170,000 | 7,405 | 5,233,329 | 2.0 | 17.61 | 46.30 |
| | Finance | FiQA-2018 | 14,166 | 648 | 57,638 | 2.6 | 10.77 | 132.32 |
| Fact Checking | Wikipedia | FEVER | 140,085 | 6,666 | 5,416,568 | 1.2 | 8.13 | 84.76 |

Table 3: Performance comparison of different approaches for Continual Document Retrieval tasks. Here, we report the nDCG@10 scores for retrieval of each task. For continually trained methods, we use the latest model after training on T5 for retrieval of all tasks. We highlight the proposed QDC-based approaches in purple. †: results excerpted from Thakur et al. (2021).

| Method | Continual | T1 - MS MARCO | T2 - NQ | T3 - Hotpot QA | T4 - FEVER | T5 - FiQA2018 | Avg. Score |
|----------------------------|-----------|---------------|---------|----------------|------------|---------------|------------|
| BM25 [†] | ✗ | 22.8 | 32.9 | 60.3 | 75.3 | 23.6 | 43.0 |
| Joint Training | ✗ | 39.2 | 50.7 | 71.7 | 75.1 | 33.8 | 54.1 |
| FT | ✓ | 34.8 | 50.4 | 59.1 | 68.2 | 34.4 | 49.4 |
| FT + QDC | ✓ | 38.4 | 52.5 | 67.5 | 75.0 | 34.4 | 53.5 |
| FT + KD | ✓ | 37.6 | 52.6 | 65.8 | 63.7 | 34.3 | 50.8 |
| FT + KD + QDC | ✓ | 39.3 | 54.1 | 69.3 | 73.6 | 34.3 | 54.1 |
| FT (with re-indexing) | ✓ | 33.8 | 45.7 | 62.9 | 73.5 | 34.4 | 50.1 |
| FT + KD (with re-indexing) | ✓ | 34.1 | 46.7 | 64.1 | 72.6 | 34.3 | 50.4 |

5 EXPERIMENTS

Datasets. We use 5 retrieval datasets from the BEIR benchmark (Thakur et al., 2021) and continually train and evaluate them in task-incremental learning setup. We use the datasets in the following sequence - MS MARCO (Bajaj et al., 2016), NQ (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), FEVER (Thorne et al., 2018) and FiQA-2018 (Maia et al., 2018). We select these datasets since they have sufficient training set of query-document pairs and hence we can continually train them. The other datasets from BEIR benchmark has very little to no training data and are primarily used for zero-shot retrieval evaluation. We discuss the details of the datasets in Table 2.

Implementation. We use the nomic embedding model (Nussbaum et al., 2024) with 768 dimensional feature embeddings for our experiments. We use the MTEB benchmark (Muennighoff et al., 2023) for evaluating the retrieval models on different tasks. We use the pre-trained model¹ from Nussbaum et al. (2024) which is pre-trained with MLM followed by unsupervised contrastive pre-training. The nomic model is a modified version of BERT base (Devlin et al., 2019) resulting in a 137M parameter model with 8192 sequence length. Starting with the pre-trained model, we fine-tune them continually for our experiments. Following Nussbaum et al. (2024), we use $\mathcal{H} = 7$ hard negatives for each query which are mined from the corresponding dataset corpus using the gte-base model² (Li et al., 2023). For each task, we train for one epoch with a batch size of 128, learning rate of 2×10^{-5} and weight decay of 0.01. Similar to Nussbaum et al. (2024), we find that training for more epochs does not improve performance. We use 4 NVIDIA L40S GPUs to train the models for our experiments. While we compare the nomic model performance with other dense retrievers in Section C, the main evaluation for the continual setting is based on the pre-trained nomic model which we continually finetune on new tasks.

Metrics. We use the Normalised Cumulative Discount Gain (nDCG@10) metric (Yining et al., 2013) for top-10 retrieved documents in our evaluations following Thakur et al. (2021); Günther et al. (2023); Nussbaum et al. (2024). We also report other metrics like Recall@10 and MAP@10 (Mean Average Precision) in Section B.

Comparison to BM25. We compare with commonly used lexical retriever BM25 (Robertson et al., 2009) in Table 3. We observe that BM25 performs very poorly on most datasets compared to FT or joint training except on FEVER where it outperforms all other approaches.

Impact of KD. We show in Table 3 that embedding knowledge distillation (KD) improves the stability of the model which is evident from better performance of old tasks where FT+KD improves MS MARCO by 2.8%, NQ by 2.2%

¹nomic pre-trained model (<https://huggingface.co/nomic-ai/nomic-embed-text-v1-unsupervised>)

²gte-base model (<https://huggingface.co/thenlper/>)

Table 4: Performance of the proposed method for Continual Document Retrieval tasks after training on each task. Here, we show the retrieval performance (nDCG@10 scores) for all datasets as the model is continually trained on a new task. We denote the performance of old tasks in yellow and the zero-shot retrieval performance of future tasks in green. The performance drop (PD) from the task in which the model is learned on a given dataset (denoted by TX) to the last task suggests the forgetting of that dataset.

| | | Training Sequence → | | | | | |
|-----------|----------|---------------------|---------|---------------|------------|---------------|--------------|
| Eval on | | T1 - MS MARCO | T2 - NQ | T3 - HotpotQA | T4 - FEVER | T5 - FiQA2018 | PD (TX - T5) |
| FT | MS MARCO | 40.2 | 38.7 | 38.2 | 36.3 | 34.8 | 5.4 |
| | NQ | 50.4 | 54.7 | 51.6 | 50.7 | 50.4 | 4.3 |
| | HotpotQA | 61.3 | 56.4 | 71.5 | 68.8 | 59.1 | 12.4 |
| | FEVER | 67.8 | 58.0 | 66.1 | 72.8 | 68.2 | 4.6 |
| | FiQA2018 | 31.4 | 32.4 | 29.4 | 29.3 | 34.4 | - |
| | Avg Acc. | 50.2 | 48.1 | 51.4 | 51.6 | 49.4 | |
| Eval on | | T1 - MS MARCO | T2 - NQ | T3 - HotpotQA | T4 - FEVER | T5 - FiQA2018 | PD (TX - T5) |
| FT+QDC | MS MARCO | 40.2 | 39.9 | 38.5 | 37.8 | 38.4 | 1.8 |
| | NQ | 50.4 | 54.7 | 53.0 | 50.8 | 52.5 | 2.2 |
| | HotpotQA | 61.3 | 56.4 | 71.5 | 70.5 | 67.5 | 4.0 |
| | FEVER | 67.8 | 58.0 | 66.1 | 72.8 | 75.0 | -2.2 |
| | FiQA2018 | 31.4 | 32.4 | 29.4 | 29.3 | 34.4 | - |
| | Avg Acc. | 50.2 | 48.3 | 51.7 | 52.2 | 53.5 | |
| Eval on | | T1 - MS MARCO | T2 - NQ | T3 - HotpotQA | T4 - FEVER | T5 - FiQA2018 | PD (TX - T5) |
| FT+KD+QDC | MS MARCO | 40.2 | 40.2 | 39.4 | 39.0 | 39.3 | 0.9 |
| | NQ | 50.4 | 54.7 | 54.1 | 52.8 | 54.1 | 0.6 |
| | HotpotQA | 61.3 | 58.7 | 72.7 | 72.5 | 69.3 | 3.4 |
| | FEVER | 67.8 | 61.1 | 67.4 | 70.5 | 73.6 | -3.1 |
| | FiQA2018 | 31.4 | 32.7 | 30.3 | 29.3 | 34.3 | - |
| | Avg Acc. | 50.2 | 49.5 | 52.8 | 52.8 | 54.1 | |

and HotpotQA by 6.7% over FT. While KD improves over FT by 1.4% on an average, we also observe that KD can affect the plasticity of the model since the performance on newer tasks are affected (FEVER drops by 4.5%).

Impact of QDC. We show in Table 3 that QDC outperforms FT and FT+KD and improves the performance of all tasks after continually training on all tasks. Using QDC improves over FT across all tasks by 4.1% on average. When used with FT+KD models, QDC improves by 3.3% on average and outperforms all other approaches. We also evaluate the impact of QDC on the performance by evaluating on all tasks after training on each task in Table 4. We observe poor performance of old tasks (in yellow) with naive fine-tuning. When using QDC for retrieval of old tasks, the performance improves for all tasks, thereby reducing the forgetting significantly (denoted by PD). Finally, using QDC with FT+KD models achieves the best average accuracy and least PD after each task. We present examples showing improved retrieval of documents using QDC in Section D. We also discuss how to use QDC in class-incremental setting in Section A.

Comparison to joint training. We compare the performance of continually trained model with the jointly trained model, which is trained on all five datasets at the same time in a static setting. We observe that the proposed method (FT+KD+QDC) performs similarly to the jointly trained model on average. Note that the joint training here considers only those hard-negatives which are mined from each dataset and are not jointly mined. In other words, we use the same hard-negatives for each dataset in both continual finetuning and joint training for fair comparison. While the joint training performance could improve with jointly mined hard-negatives, we follow the standard practice of joint training (Nussbaum et al., 2024).

Generalizability. We also evaluate the zero-shot performance in Table 4 (in green) for unseen tasks and demonstrate how training on each new task affects the generalizability of the model. The zero-shot performance depends on the latest task the model is trained on. For instance, we see a drop in performance of FEVER after training on NQ while it improves after training on Hotpot QA. This could be due to high domain overlap between Hotpot QA and FEVER as shown in (Thakur et al., 2021). We also observe an improvement in zero-shot performance when using KD.

Comparison to re-indexing. While re-indexing was found to be effective in image retrieval (Shen et al., 2020), we show that in CDR, re-indexing of old task documents does not improve performance significantly and performs worse compared to FT on initial tasks like MS MARCO and NQ. The proposed method QDC significantly outperforms

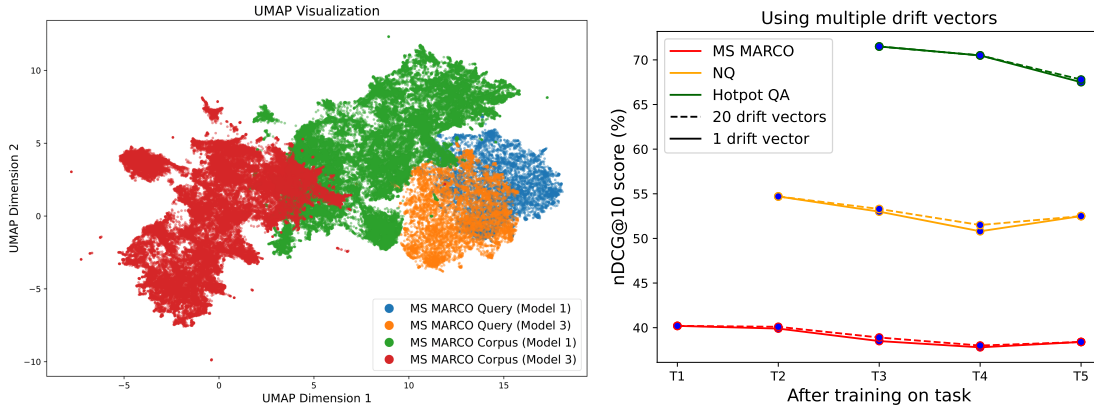


Figure 3: (left) UMAP visualization of the drift in query and corpus embeddings of MS MARCO after fine-tuning on NQ and Hotpot QA (using Model after task 3). (right) Analysis of the performance when using multiple drift vectors to represent drift between embedding spaces of old and new task.

re-indexing approach, even with re-indexing on the model trained with embedding distillation as shown in Table 3. Previously, Wang et al. observed unequal drifts in image and text modalities for continual multimodal retrieval. We observe a similar phenomenon here and show in Fig. 3 (left) that the drift in corpus document embeddings is higher compared to the drift in query embeddings for MS MARCO after training on NQ followed by Hotpot QA. Thus, the corpus document embeddings (in red) are poorly aligned with the query embeddings (in orange) in the updated embedding space after task 3.

We attribute the poor performance of re-indexing in CDR to the unequal drift in query and document embeddings. We hypothesize that the unequal drift could be due to the difference in length of queries and documents. Queries are much shorter in length while documents could be a paragraph. From Table 2, we observe that documents have a much higher average word lengths than queries (MS MARCO documents are 10 times longer than the queries on an average). This suggests that the documents could face higher forgetting or higher embedding drift compared to queries.

In the case of multi-modal retrieval (Wang et al., 2021), the unequal drift arises from the modality gap at initialization which is preserved during the contrastive training approach (as discussed in Liang et al. (2022) for vision-language models). A recent work (Schrodi et al., 2025) extensively analyzed and presented that the gap between modalities actually arises from the information imbalance in the two modalities where one modality (visual) has more information than the other one (text). We think that this explanation could also be applicable to our setting despite that we only have a single modality, since there is a significant information imbalance between queries and corpus. The imbalance in information is caused by the difference in the lengths of queries and corpus where queries are usually much shorter in length and the corpus are much longer and more detailed.

We analyze in Table 5 how the queries and corpus of different lengths of the first task MS Marco drift after training on NQ. We divide the queries and corpus into three groups (short, medium and long) based on lengths and report the average of the cosine drift in each group. We observe that corpus documents having more information drift more than the queries as also seen in Fig. 3 (left). We observe that within the queries, the medium-sized and longer queries drift more on average than the shorter ones. Similarly, among the corpus documents, we see that longer documents drift more than the short ones. This confirms the correlation between the length of the query or document and the drift.

Table 5: Average drift values for Query and Corpus across text lengths.

| | Short | Medium | Long |
|---------------|--------|--------|--------|
| Query | 0.0340 | 0.0375 | 0.0393 |
| Corpus | 0.0559 | 0.0573 | 0.0635 |

Enabling compatibility with QDC. There are two ways of maintaining query and corpus compatibility by keeping them in the same embedding space, one by re-indexing (bringing documents forward to the new space) and the other by projecting queries back to the old embedding space. While re-indexing solves the embedding space alignment problem and brings query and corpus in the same space, it does not maintain good discriminative power for the old task corpus documents since it uses the updated model which is not the best model for the old task. On the other hand, we preserve the discriminative capabilities of the model to encode the corpus in QDC since we use the documents indexed with the old model as shown in Table 6.

So, despite adding more computation costs, FT with re-indexing uses the updated model with reduced discriminative capabilities for both queries and the corpus of the old task. While QDC still uses the best model for the old task

Table 6: Comparison of methods and their properties across tasks.

| Task | Method | Query Embedding | Corpus Embedding | Compatibility | Discriminative Power for T1 |
|------|---------------------|-----------------|------------------|---------------|-----------------------------|
| T1 | FT | T1 | T1 | ✓ | ✓ |
| T2 | FT | T2 | T1 | ✗ | ✓ |
| T2 | FT with re-indexing | T2 | T2 | ✓ | ✗ |
| T2 | FT+QDC | T1 | T1 | ✓ | ✓ |

for indexing the documents and the drift compensated queries solves the alignment problem. Note that the corpus embeddings play a more important role here since a single query embedding is searched across millions of corpus documents. So, it is more important to preserve the corpus embedding space for old tasks and the best way is to use the old model indexed embeddings.

QDC with multiple drift vectors. In the proposed method, we consider a single drift vector for each task transition. One could also estimate multiple drift vectors in the embedding space. This could be done by dividing the embedding space into k clusters with corresponding centroids P_k and estimating a drift vector for each of these centroids $\Delta_k^{t-1 \rightarrow t}$ by averaging the drift of queries belonging to each cluster. During retrieval, the queries $q_{t'}$ could be assigned to the closest centroid k' and then compensated with the drift vector of the centroid $\Delta_{k'}^{t' \rightarrow t}$ as follows:

$$k' = \arg \max_k S(P_k, f_t(q_{t'})); \quad \hat{f}_{t'}(q_{t'}) = f_t(q_{t'}) - \Delta_{k'}^{t' \rightarrow t} \quad (8)$$

where k' is the closest cluster to the query embedding. Using multiple drift vectors involves storing the cluster centroids and the drift vectors for each centroid.

In our experiments in Fig. 3 (right), we empirically demonstrate that using a single drift vector is effective to estimate the query drift and estimating multiple drift vectors for each task transition does not improve the performance significantly. Using 20 drift vectors improves the performance of MS MARCO by 0.4% after T3, NQ by 0.7% after T4, Hotpot QA by 0.3% after T5, while achieving the same performance for FEVER. Based on these observations, we advocate using a single drift vector for each task which achieves similar performance with simpler and faster retrieval since it does not need to find the closest cluster centroid for drift compensation.

6 CONCLUSION

In this work, we study how continually training embedding models on query-document pairs from new datasets over time could affect the retrieval performance across all seen tasks. We observe forgetting on old tasks in CDR and show that using knowledge distillation on the query and document embeddings can reduce the forgetting. We discuss the issue of non-compatibility between query and indexed corpus embeddings due to *embedding drift* after the embedding model is updated on a new task. We propose a novel method to avoid this issue by estimating the drift of queries from old to new embedding space and then compensating the estimated drift to project the queries to old embedding space at test time. This enables compatibility since the indexed embeddings were extracted from the old model and thus we compute similarities for retrieval with queries and corpus in the same embedding space. We establish a continual training benchmark with five large-scale datasets and demonstrate that the proposed QDC approach outperforms other approaches. We also show that re-indexing based approach does not perform well despite being very expensive.

We believe that enabling compatibility in continually trained retrieval embedding models will benefit several practical document retrieval applications like RAG systems where the retriever embedding model could be continually updated to add new knowledge over time. We hope that our approach and findings will encourage further research and more extensive benchmarks on continual document retrieval.

Acknowledgements. We acknowledge projects PID2022-143257NB-I00, financed by MCIN/AEI/10.13039/501100011033 and FSE+, funding by the European Union ELLIOT project, and the Generalitat de Catalunya CERCA Program. Dipam Goswami acknowledges travel support from ELISE (GA no 951847). Bartłomiej Twardowski acknowledges the grant RYC2021-032765-I and National Centre of Science (NCN, Poland) Grant No. 2023/51/D/ST6/02846. Liying Wang acknowledges financial support from China Scholarship Council.

REFERENCES

- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.
- Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *International Conference on Computer Vision (ICCV)*, 2019.
- Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 192–199, 2000.
- Niccolò Biondi, Federico Pernici, Matteo Bruni, Daniele Mugnai, and Alberto Del Bimbo. Cl2r: Compatible lifelong learning representations. *ACM Transactions on Multimedia Computing, Communications and Applications*, 18(2s): 1–22, 2023.
- Niccolò Biondi, Federico Pernici, Simone Ricci, and Alberto Del Bimbo. Stationary representations: Optimally approximating compatibility and implications for improved model replacements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28793–28804, 2024.
- Jorg Bornschein, Alexandre Galashov, Ross Hemsley, Amal Rannen-Triki, Yutian Chen, Arslan Chaudhry, Xu Owen He, Arthur Douillard, Massimo Caccia, Qixuan Feng, et al. Nevis’ 22: A stream of 100 tasks sampled from 30 years of computer vision research. *Journal of Machine Learning Research*, 24(308):1–77, 2023.
- Yinqiong Cai, Keping Bi, Yixing Fan, Jiafeng Guo, Wei Chen, and Xueqi Cheng. L2r: Lifelong learning for first-stage retrieval with backward-compatible representations. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 183–192, 2023.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder for english. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pp. 169–174, 2018.
- Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2021.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, volume 1 (long and short papers)*, pp. 4171–4186, 2019.
- Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision (ECCV)*, 2020.
- Thomas Gerald and Laure Soulier. Continual learning of long topic sequences in neural information retrieval. In *Advances in Information Retrieval*, pp. 244–259, 2022.
- Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. End-to-end retrieval in continuous space. *arXiv preprint arXiv:1811.08008*, 2018.
- Alex Gomez-Villa, Dipam Goswami, Kai Wang, Andrew D. Bagdanov, Bartłomiej Twardowski, and Joost van de Weijer. Exemplar-free continual representation learning via learnable drift compensation. In *Computer Vision – ECCV 2024*, pp. 473–490, 2025.
- Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost van de Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- Dipam Goswami, Albin Soutif-Cormerais, Yuyang Liu, Sandesh Kamath, Bartłomiej Twardowski, and Joost van de Weijer. Resurrecting old classes with new data for exemplar-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28525–28534, 2024.

- Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdesslem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, et al. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *arXiv preprint arXiv:2310.19923*, 2023.
- Jingrui Hou, Georgina Cosma, and Axel Finke. Advancing continual lifelong learning in neural information retrieval: definition, dataset, framework, and empirical evaluation. *Information Sciences*, 687:121368, 2025.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.
- Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43, 2023.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen Tau Yih. Dense passage retrieval for open-domain question answering. In *2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pp. 6769–6781. Association for Computational Linguistics (ACL), 2020.
- Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.
- Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 39–48, 2020.
- Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. Open-world continual learning: Unifying novelty detection and continual learning. *Artificial Intelligence*, 338:104237, 2025.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences (PNAS)*, 2017.
- Varsha Kishore, Chao Wan, Justin Lovelace, Yoav Artzi, and Kilian Q Weinberger. Incdsi: Incrementally updatable document retrieval. In *International conference on machine learning*, pp. 17122–17134. PMLR, 2023.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 2019.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.
- Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.
- Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.
- Davis Liang, Peng Xu, Siamak Shakeri, Cicero Nogueira dos Santos, Ramesh Nallapati, Zhiheng Huang, and Bing Xiang. Embedding-based zero-shot retrieval through query generation. *arXiv preprint arXiv:2009.10270*, 2020.
- Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Advances in Neural Information Processing Systems*, 35:17612–17625, 2022.
- Jesús Lovón-Melgarejo, Laure Soulier, Karen Pinel-Sauvagnat, and Lynda Tamine. Studying catastrophic forgetting in neural ranking models. In *Advances in Information Retrieval*, pp. 375–390, 2021.
- Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345, 2021.

- Simone Magistri, Tomaso Trinci, Albin Soutif, Joost van de Weijer, and Andrew D Bagdanov. Elastic feature consolidation for cold start exemplar-free incremental learning. In *The Twelfth International Conference on Learning Representations*, 2024.
- Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. Www’18 open challenge: financial opinion mining and question answering. In *Companion proceedings of the the web conference 2018*, pp. 1941–1942, 2018.
- Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.
- Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2022.
- Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Elsevier, 1989.
- Sanket Vaibhav Mehta, Jai Gupta, Yi Tay, Mostafa Dehghani, Vinh Q Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler. Dsi++: Updating transformer memory with new documents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8198–8213, 2023.
- Niklas Muennighoff. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*, 2022.
- Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, 2023.
- Zach Nussbaum, John X Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*, 2024.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *Winter Conference on Applications of Computer Vision (WACV)*, 2023.
- Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023.
- Vivek Ramanujan, Pavan Kumar Anasosalu Vasu, Ali Farhadi, Oncel Tuzel, and Hadi Pouransari. Forward compatible training for large-scale embedding retrieval systems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19386–19395, 2022.
- Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.
- Simon Schrodi, David T. Hoffmann, Max Argus, Volker Fischer, and Thomas Brox. Two effects, one trigger: On the modality gap, object bias, and information imbalance in contrastive vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=uAFHCZRmXk>.
- Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pp. 4548–4557. PMLR, 2018.
- Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. Towards backward-compatible representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6368–6377, 2020.

- James Seale Smith, Junjiao Tian, Shaunak Halbe, Yen-Chang Hsu, and Zsolt Kira. A closer look at rehearsal-free continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2410–2420, 2023.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.
- James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 809–819, 2018.
- Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, and Aidan N Gomez. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008, 2017.
- Eli Verwimp, Rahaf Aljundi, Shai Ben-David, Matthias Bethge, Andrea Cossu, Alexander Gepperth, Tyler L Hayes, Eyke Hüllermeier, Christopher Kanan, Dhireesha Kudithipudi, et al. Continual learning: Applications and the road forward. *Transactions on Machine Learning Research*, 2024.
- Timmy ST Wan, Jun-Cheng Chen, Tzer-Yi Wu, and Chu-Song Chen. Continual learning for visual search with backward consistent feature embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16702–16711, 2022.
- Kai Wang, Luis Herranz, and Joost van de Weijer. Continual learning in cross-modal retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3628–3638, 2021.
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.
- Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.
- Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. Pretrained transformers for text ranking: Bert and beyond. In *Proceedings of the 14th ACM International Conference on web search and data mining*, pp. 1154–1156, 2021.
- Wang Yining, Wang Liwei, Li Yuanzhi, He Di, Chen Wei, and Liu Tie-Yan. A theoretical analysis of ndcg ranking measures. In *Proceedings of the 26th annual conference on learning theory*, 2013.
- Lu Yu, Bartłomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

A USING QDC IN CLASS-INCREMENTAL LEARNING SETTING

It would be interesting to extend QDC to CIL setting with no access to task-id and future works could explore that setting. Similar to several existing works in CIL which predicts task-id (see discussion in Kim et al. (2025)), the proposed method QDC could be adapted to CIL setting by predicting the task-id of a given query as discussed below.

- After training on each task, the feature centroid of that task can be stored. Centroids of all old tasks could be updated by adding the drift vector $\Delta^{t-1 \rightarrow t}$ of the current task. So, at the end of task t , we have the task centroids for all tasks in the updated embedding space.
- During inference with queries extracted using the latest task model, we can predict the task-id based on cosine distance between query embedding and the task centroids.
- After predicting the task-id for queries, we can perform QDC to move the query back to the embedding space of the task.

B PERFORMANCE EVALUATION WITH OTHER METRICS

We show the performance of different methods using other metrics like recall@10 and MAP@10 in Tables 7 and 8. We observe the same trend that using QDC outperforms FT and FT+KD and achieves similar performance as joint training.

Table 7: Performance comparison of different approaches for Continual Document Retrieval tasks. Here, we report the **recall@10** scores for retrieval of each task. For continually trained methods, we use the latest model after training on T5 for retrieval of all tasks. We highlight the proposed QDC-based approaches in purple .

| Method | T1 - MS MARCO | T2 - NQ | T3 - Hotpot QA | T4 - FEVER | T5 - FiQA2018 | Average |
|----------------------------|---------------|---------|----------------|------------|---------------|-------------|
| Joint Training | 60.8 | 72.8 | 75.5 | 87.9 | 40.3 | 67.5 |
| FT | 54.2 | 71.3 | 64.5 | 81.9 | 41.1 | 62.6 |
| FT + QDC | 59.5 | 73.3 | 72.2 | 87.1 | 41.1 | 66.6 |
| FT + KD | 58.2 | 73.3 | 70.6 | 78.2 | 41.4 | 64.3 |
| FT + KD + QDC | 60.6 | 75.3 | 73.7 | 86.1 | 41.4 | 67.4 |
| FT (with re-indexing) | 53.6 | 67.0 | 67.6 | 87.4 | 41.1 | 63.3 |
| FT + KD (with re-indexing) | 53.8 | 68.1 | 69.2 | 86.1 | 41.4 | 63.7 |

Table 8: Performance comparison of different approaches for Continual Document Retrieval tasks. Here, we report the **MAP@10** scores for retrieval of each task. For continually trained methods, we use the latest model after training on T5 for retrieval of all tasks. We highlight the proposed QDC-based approaches in purple .

| Method | T1 - MS MARCO | T2 - NQ | T3 - Hotpot QA | T4 - FEVER | T5 - FiQA2018 | Average |
|----------------------------|---------------|---------|----------------|------------|---------------|-------------|
| Joint Training | 32.2 | 42.6 | 63.9 | 69.7 | 26.5 | 47.0 |
| FT | 28.6 | 42.8 | 50.6 | 62.7 | 27.0 | 42.3 |
| FT + QDC | 31.6 | 44.8 | 59.3 | 69.9 | 27.0 | 46.5 |
| FT + KD | 30.9 | 44.9 | 57.4 | 58.1 | 26.9 | 43.6 |
| FT + KD + QDC | 32.4 | 46.2 | 61.2 | 68.5 | 26.9 | 47.0 |
| FT (with re-indexing) | 27.4 | 37.8 | 54.9 | 67.9 | 27.0 | 43.0 |
| FT + KD (with re-indexing) | 27.7 | 38.9 | 56.0 | 67.1 | 26.9 | 43.3 |

C COMPARISON OF NOMIC PRE-TRAINED MODEL WITH OTHER RETRIEVAL MODELS

We compare the performance of the nomic retriever model trained either jointly or continually with classical dense retrievers like DPR (Karpukhin et al., 2020), ColBERT (Khattab & Zaharia, 2020) and ANCE (Xiong et al., 2021). We report performance of DPR, ColBERT and ANCE from BEIR (Thakur et al., 2021). While the other dense retriever models like ColBERT perform competitively, the nomic retriever model outperforms them. For the continual setting, we base the comparison on the nomic model. Benchmarking the performance of these other retriever architectures by continually training them in the proposed continual setting could be interesting to explore in future works.

Table 9: Performance comparison of different approaches. We report the nDCG@10 scores for retrieval of each task. Here, we do not consider continual training and evaluate the performance on each task separately. †: results excerpted from [Thakur et al. \(2021\)](#).

| Method | Continual | T1 - MS MARCO | T2 - NQ | T3 - Hotpot QA | T4 - FEVER | T5 - FiQA2018 | Avg. Score |
|------------------------|-----------|---------------|---------|----------------|------------|---------------|-------------|
| BM25† | ✗ | 22.8 | 32.9 | 60.3 | 75.3 | 23.6 | 43.0 |
| DPR† | ✗ | 17.7 | 47.4 | 39.1 | 56.2 | 11.2 | 34.3 |
| ANCE† | ✗ | 38.8 | 44.6 | 45.6 | 66.9 | 29.5 | 45.1 |
| ColBERT† | ✗ | 40.1 | 52.4 | 59.3 | 77.1 | 31.7 | 52.1 |
| Joint Training (Nomic) | ✗ | 39.2 | 50.7 | 71.7 | 75.1 | 33.8 | 54.1 |
| FT + KD + QDC (Nomic) | ✓ | 39.3 | 54.1 | 69.3 | 73.6 | 34.3 | 54.1 |

D RETRIEVAL EXAMPLES

We present some retrieval results using the continually fine-tuned model in Tables 10 and 11 for old tasks (MS MARCO and Hotpot QA). We show that using the proposed QDC method with fine-tuned model retrieves more relevant documents from the corpus for a given query.

Table 10: Examples of top-1 retrieved document for a given query using fine-tuned model and with the proposed QDC (FT+QDC) for **MS MARCO** dataset using the continually trained model after task 5.

| Query | Document retrieved with FT | Document retrieved with FT+QDC |
|--------------------------------------|--|--|
| do vhi swiftcare do blood tests? | Blood tests. Information on having blood tests and the types of blood tests you might have. Your blood sample is sent to the laboratory. A blood doctor can look at your sample under a microscope. They can see the different types of cells and can count the different blood cells. | The SwiftCare clinics charge an initial consultation fee of 85 euro, with additional charges for tests and procedures. For example, an x-ray at the clinics costs 65 euro, blood tests range from 30 to 50 euro and complex suturing costs 50 euro. The Swiftcare clinics are staffed by doctors with significant experience in general practice and emergency care, according to VHI. The clinics are run as a joint initiative between the VHI and The Well, a primary care medical company. |
| the miners state bank routing number | Search all THE MINERS STATE BANK routing numbers in the table below. Use the Search box to filter by city, state, address, routing number. Click on the routing number link in the table below to navigate to it and see all the information about it (address, telephone number, zip code, etc.). | The Miners State Bank Routing Number the miners state bank routing aba number 091109253 routing number is a 9-digit number designed and assigned to The Miners State Bank by The American Bankers Association (ABA) to identify the financial institution upon which a payment was drawn. |
| cadillac alternator price | 1 On average, a car alternator prices are going to range anywhere from 66to320. 2 This is not going to include the labor costs. 3 When you factor in labor costs, itâ safe to add another 100to275. | A Cadillac De Ville Alternator Replacement costs between 378and860 on average. Get a free detailed estimate for a repair in your area. A Cadillac De Ville Alternator Replacement costs between 378and860 on average. |
| door knocker definition | Definition of 'knocker'. knocker. A knocker is a piece of metal on the front door of a building, which you use to hit the door in order to attract the attention of the people inside. | Door Knocker definition. An act of physical violence performed on a person (usually a woman) who is wearing huge hoop earrings. A cob of dried corn employed by pranksters on Mischief evening to toss at people's doors.....notable for loud BLANG!! or KABAAM!! Not what title really seems. |

Table 11: Examples of top-1 retrieved document for a given query using fine-tuned model and with the proposed QDC (FT+QDC) for **Hotpot QA** dataset using the continually trained model after task 5.

| Query | Document retrieved with FT | Document retrieved with FT+QDC |
|---|---|---|
| This singer of A Rather Blustery Day also voiced what hedgehog? | Pinocchio (singer) Pinocchio is a fictional, animated French character and singer. | A Rather Blustery Day A Rather Blustery Day is a whimsical song from the Walt Disney musical film featurette, Winnie the Pooh and the Blustery Day. It was written by Robert & Richard Sherman and sung by Jim Cummings as Pooh. |
| What WB supernatural drama series was Jawbreaker star Rose McGowan best known for being in? | Charmed (season 4) The fourth season of Charmed, an American supernatural drama television series, began airing on October 4, 2001 on The WB. Airing on Thursdays at 9:00 pm, the season consisted of 22 episodes and concluded its airing on May 16, 2002. This season also saw the introduction of Rose McGowan as Paige Matthews—half-sister to Prue, Piper and Phoebe—and a slight alteration of the opening credits, due to the third season departure of Shannen Doherty as Prue. Paramount Home Entertainment released the complete fourth season in a six-disc boxed set on February 28, 2006. | Rose McGowan Rose Arianna McGowan (born September 5, 1973) is an Italian-born American actress, film producer, director and singer. She is best known to television audiences for having played Paige Matthews in The WB supernatural drama series Charmed from 2001 to 2006. |
| In which role did Caroline Carver played in a 1999 Hallmark Entertainment made-for-TV fantasy movie? | The Magical Legend of the Leprechauns The Magical Legend of the Leprechauns is a 1999 Hallmark Entertainment made-for-TV fantasy movie. It stars Randy Quaid, Colm Meaney, Kieran Culkin, Roger Daltrey, Caroline and Whoopi Goldberg. The film contains two main stories that eventually Carver intertwine: the first being the story of an American businessman who visits Ireland and encounters magical leprechauns and the second, a story of a pair of star-crossed lovers who happen to be a fairy and a leprechaun, belonging to opposing sides of a magical war. It contains many references to Romeo and Juliet such as two lovers taking poison and feuding clans. | Caroline Carver (actress) Caroline Carver (born 1976) is an English actress, screenwriter, and producer best known for roles such as Princess Jessica in the TV film The Magical Legend of the Leprechauns (1999), Ingrid in The Aryan Couple (2004), and Sandy in My First Wedding (2006). |
| What's the name of the fantasy film starring Sarah Bolger, featuring a New England family who discover magical creatures around their estate? | Fredegar Bolger Fredegar Fatty Bolger is a fictional character in J. R. R. Tolkien's fantasy novel The Lord of the Rings. | Sarah Bolger Sarah Lee Bolger (born 28 February 1991) is an Irish actress. She is best known for her roles in the films In America, Stormbreaker, and The Spiderwick Chronicles, as well as her award-winning role as Lady Mary Tudor in the TV series The Tudors, and for guest starring as Princess Aurora in Once Upon a Time. |