L^{2}R: Lifelong Learning for First-stage Retrieval with Backward-Compatible Representations

Yinqiong Cai
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
caiyinqiong18s@ict.ac.cn

Keping Bi
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
bikeping@ict.ac.cn

Yixing Fan
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
fanyixing@ict.ac.cn

Jiafeng Guo*
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
guojiafeng@ict.ac.cn

Wei Chen
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
chenwei2022@ict.ac.cn

Xueqi Cheng
CAS Key Lab of Network Data Science and Technology, ICT, CAS
University of Chinese Academy of Sciences
Beijing, China
cxq@ict.ac.cn

ABSTRACT

First-stage retrieval is a critical task that aims to retrieve relevant document candidates from a large-scale collection. While existing retrieval models have achieved impressive performance, they are mostly studied on static data sets, ignoring that in the real-world, the data on the Web is continuously growing with potential distribution drift. Consequently, retrievers trained on static old data may not suit new-coming data well and inevitably produce sub-optimal results. In this work, we study lifelong learning for first-stage retrieval, especially focusing on the setting where the emerging documents are unlabeled since relevance annotation is expensive and may not keep up with data emergence. Under this setting, we aim to develop model updating with two goals: (1) to effectively adapt to the evolving distribution with the unlabeled new-coming data, and (2) to avoid re-inferring all embeddings of old documents to efficiently update the index each time the model is updated.

We first formalize the task and then propose a novel Lifelong Learning method for the first-stage Retrieval, namely L^{2}R. L^{2}R adopts the typical memory mechanism for lifelong learning, and incorporates two crucial components: (1) selecting *diverse support negatives* for model training and memory updating for effective model adaptation, and (2) a *ranking alignment objective* to ensure the backward-compatibility of representations to save the cost of index rebuilding without hurting the model performance. For evaluation, we construct two new benchmarks from LoTTE and Multi-CPR datasets to simulate the document distribution drift in realistic retrieval scenarios. Extensive experiments show that L^{2}R significantly outperforms competitive lifelong learning baselines.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**.

KEYWORDS

Neural Retrieval Models, Lifelong Learning, Distribution Drifts

ACM Reference Format:
Yinqiong Cai, Keping Bi, Yixing Fan, Jiafeng Guo, Wei Chen, and Xueqi Cheng. 2023. L^{2}R: Lifelong Learning for First-stage Retrieval with Backward-Compatible Representations. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614947

1 INTRODUCTION

First-stage retrieval aims to quickly retrieve a few relevant document candidates from a large-scale collection, which has become a core component in information retrieval (IR) applications [12, 47]. While retrieval models based on pre-trained language models (PLMs) [21, 36, 45, 48] have demonstrated impressive performance, most of them are studied on static datasets, neglecting that in the real world, new documents are continuously emerging on the Web. For example, when a new event (e.g., ChatGPT) breaks out, a large number of documents on this topic were generated and shared, accompanied by booming information needs regarding the topic (searching for not only new documents but also old ones). The emerging documents and queries on the new topics may cause the distribution of retrieval collection to drift over time. Consequently, directly applying the model trained on previous data to the new collection is obviously not an optimal solution. Then, how can we continuously learn a retrieval model to adapt to the evolving data distribution effectively and efficiently? To study this problem, we formalize the task of lifelong learning for first-stage retrieval.

*Jiafeng Guo is the corresponding author.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3614947

Lifelong learning [6, 41] has been widely studied in the machine learning community, especially on computer vision (CV) tasks [25, 43]. In a typical setting of lifelong learning, a model is set to learn with non-identically and independently distributed (non-I.I.D.) new-coming data [43], with the goal of preserving acquired knowledge and learning new knowledge. Most research [1, 18, 35] in this field focuses on addressing the catastrophic forgetting issue [11, 26], i.e., the model's inability to perform well on previously seen data after being updated with new data. One representative paradigm for lifelong learning is the memory-based method [1, 5], which stores and replays historical samples while training on new data to mitigate the forgetting of acquired knowledge. These lifelong learning methods have been shown to be effective in various CV tasks [30, 46]. However, there has been limited research on the lifelong learning problem for IR tasks.

In this paper, we study the task of lifelong learning for first-stage retrieval in a setting where new documents are unlabeled. We focus on this setting in our initial attempt because relevance annotation on the new data is expensive and may not catch up with data emergence. In this setting, besides the essential goal of general lifelong learning, i.e., preserving acquired knowledge and learning new knowledge, it poses several new challenges:

1) Without labeled positive samples, new data could have limited benefit to supervise model learning. Moreover, the unlabeled positives in new data could mislead the model if we simply take all new documents as irrelevant for training.
2) It incurs significant costs to re-compute all document representations and rebuild the entire index each time the model is updated. It would be ideal to avoid repeated representation computation without harming model performance.
3) The pairwise modeling of query-document pairs in IR makes the task more complicated, compared to the pointwise modeling of the classification tasks in CV. For any query, either seen or unseen, the model needs to achieve good retrieval performance on both new and old documents.

Due to these challenges, existing lifelong learning methods in other fields cannot be directly applied to the retrieval task. Although some work has explored the catastrophic forgetting issue of re-ranking models under the lifelong learning setting, no feasible solutions are proposed to solve it [10, 23].

To address the above challenges, we propose a memory-based Lifelong Learning method for first-stage Retrieval, named as L^{2}R. L^{2}R maintains a buffer to store the historical support negatives (i.e., negative samples that are important for learning the decision boundary of the model) for each training query, and when a session of unlabeled new documents arrives, it updates the model as follows: 1) To adapt the model to the new distribution, L^{2}R selects diverse support negatives from the unlabeled new data for model training, by estimating their confidence of being hard negatives and redundancy with other selected ones. 2) To balance the model's ability to preserve acquired knowledge and learn new knowledge, L^{2}R selects historical support documents distinct from the selected new samples and uses them together for model updating. 3) To avoid re-inferring embeddings of old documents each time the model is updated, L^{2}R incorporates a novel ranking alignment objective to ensure the backward compatibility of document representations without harming retrieval performance. Overall, through the selection strategy of *diverse support negatives* and the *ranking alignment objective* for compatible learning, L^{2}R enables effective and efficient retrieval model lifelong learning.

For evaluation, we construct two benchmarks based on the LoTTE [36] and Multi-CPR [22] datasets, namely LL-LoTTE and LL-MultiCPR, to simulate the realistic retrieval scenario where documents emerge continuously with distribution drift. The empirical results on both benchmarks show that L^{2}R outperforms representative and state-of-the-art lifelong learning baselines in terms of metrics on both learning new data and addressing the forgetting issue. Moreover, our proposed ranking alignment objective achieves not only representation backward compatibility but also remarkably even better performance. We further confirm the advantages of our model through in-depth studies on the data selection strategy and the backward-compatible alignment objectives.

2 RELATED WORK

Lifelong Learning. Lifelong learning [6], also referred to as continual learning [41] or incremental learning [4], has received much attention in building adaptive systems that are able to gain, retain, and transfer knowledge when facing non-stationary data streams. Research in this field mainly focuses on solving the catastrophic forgetting issue [25, 43]. There are three main method paradigms, including regularization-based [16, 18], architecture-based [7, 35] and memory-based methods [1, 5, 17, 32].

Lifelong learning has been widely studied in various machine learning tasks [30, 32, 39, 46]. Recently, Mai et al. [25] surveyed a wide range of methods to address the lifelong learning problem for image classification. In natural language processing tasks, the research on lifelong learning mostly focuses on pre-training [3, 44]. For example, Qin et al. [29] proposed ELLE for incremental pre-training on emerging data efficiently. Wu et al. [44] compared the performance over the combination of five PLMs and four lifelong learning approaches. However, to our best knowledge, there have been no studies on lifelong learning for first-stage retrieval.

First-stage Retrieval. In recent years, substantial efforts have been made on various retrieval models [12, 47], including both classical term-based methods like BM25 [34], and more recent PLMs-based dense retrieval models [9, 19, 20, 24]. PLMs-based retrieval models have compelling performance and are widely adopted in the industry. However, most existing studies are on static document sets, ignoring the realistic scenario wherein new documents continually arrive at the system.

Lifelong learning for information retrieval (IR) is an important but less-explored topic, including both the first-stage retrieval and re-ranking stage. Recently, Lovón-Melgarejo et al. [23] and Gerald and Soulier [10] studied the lifelong learning problem for PLMs-based re-ranking models. They observed the catastrophic forgetting issue in lifelong IR model learning. Later, Mehta et al. [27] studied continual learning for generative IR models [40], in which they studied how to incrementally index new documents into the model parameters, instead of the distribution shift caused by newly emerged data. Lifelong learning has been studied in image retrieval [37, 42]. However, the experiments were conducted on fine-grained image classification datasets, the settings of which completely differ from the realistic scenario for document retrieval.

Compatible Representation Learning. Learning compatible representations [8, 13, 31, 37] is a practical need in many scenarios, with the goal of ensuring the embeddings generated by different models are compatible. For example, BCT [37] and LCE [28] learn compatible representations for image recognition, where the embeddings computed by the updated model are directly comparable to those generated by previous models. Specifically, BCT [37] constrains the feature space by simultaneously enabling gradient flow from both the old and new classifiers. However, this method is not suitable for first-stage retrieval, since the relevance score is calculated on the embeddings directly and there are no classification layers. LCE [28] bridges the multiple feature spaces via a lightweight transformation function. However, they still need to re-compute all embeddings of the previous images, which is inefficient for large collection in retrieval. Beyond these, representation compatibility has received increasing attention in asymmetric retrieval [8], where the query and document use different models due to the constrained resources of the computing platform. In contrast to these methods, we study compatible representation learning under the lifelong learning setting for first-stage retrieval.

3 TASK DESCRIPTION

First-stage Retrieval. Given a query q and a document collection $\mathcal{D}_0$, first-stage retrieval aims to find potentially relevant documents. With a labeled training dataset $C_0 = \{(q, D_q^+)\}$, where q is a query and $d_q^+ \in D_q^+$ is one of the relevant documents for q, we can build an initial retrieval model f_0 using a dual-encoder architecture with a standard contrastive learning objective [12, 47]. Then, the embeddings of documents in $\mathcal{D}_0$ are extracted and indexed, and the retrieval is performed by estimating the similarity between the query embedding with the document embeddings in the index.

Lifelong Learning for First-stage Retrieval. A stream of document sets $\{\mathcal{D}_1, \cdots, \mathcal{D}_T\}$ having different distributions arrive in T sessions sequentially. Note that these new documents have no relevance labels. For any session $t \in \{1, \cdots, T\}$, the lifelong learning algorithm $\mathcal{A}$ utilizes documents in $\mathcal{D}_t$ to update f_{t-1} to f_t, aiming to adapt the retriever to the new distribution,

$$\mathcal{A}_t : \langle f_{t-1}, C_0, \mathcal{D}_t, M_{t-1} \rangle \rightarrow \langle f_t, M_t \rangle, \tag{1}$$

where M_{t-1} and M_t are the external memory for session $t-1$ and t respectively, which store useful information for lifelong model learning, e.g., a subset of training samples or historical versions of the model. If the model updating is representation backward-compatible, at session t, we only need to compute document embeddings for $\mathcal{D}_t$ using model f_t. The embeddings of $\mathcal{D}_{0:t-1} = \bigcup_{i=0}^{t-1} \mathcal{D}_i$ that are computed with historical models can be reused when updating the index with existing techniques [14]. Otherwise, we need to use f_t to compute the embeddings for all documents in $\mathcal{D}_{0:t}$ to rebuild the retrieval index.

4 METHODOLOGY

4.1 Overview of the Approach

We employ the typical memory mechanism [5, 38] in L^{2}R and maintain a restricted external memory to store a subset of historical documents for each training query. This memory mechanism enables the model to efficiently determine the replay samples to address the catastrophic forgetting issue, without browsing from the entire collection. Based on the memory mechanism, for effective and efficient lifelong learning of retrieval models, L^{2}R incorporates two important components, including the selection strategy of **diverse support negatives** and the **ranking alignment objective** for backward-compatible representation learning.

Algorithm 1: Overview of L^{2}R.

Input: Dataset C_0, Retrieval model f_0, Memory buffer M_0 with n slots for each query, Total sessions T
Output: Retrieval Model f_T

```
for t ∈ {1, ··· , T} do
    f_t ← f_{t−1}; M_c ← {};
    for (q, d_q^+) ~ C_0 do
        D_q^new ← NewDataSelection(q, d_q^+, D_t);
        M_c ← M_c ∪ D_q^new;
        D_q^mem ← MemoryDataSelection(q, D_q^new, M_{t−1});
        f_t ← ModelUpdate(q, d_q^+, D_q^new ∪ D_q^mem, f_t);
    end
    M_t ← MemoryUpdate (q, M_c, M_{t−1}, n);
end
return f_T
```

As shown in Algorithm 1, when the newly emerged data $\mathcal{D}_t$ arrives at session t, L^{2}R selects diverse support negative documents from the new data and the memory buffer M_{t-1} respectively, then updates the retriever from f_{t-1} to f_t with the selected samples, and finally updates the memory with the new data. Next, we introduce the detailed data selection method (Section 4.2), and the optimization objective for compatible learning (Section 4.3).

4.2 Diverse Support Negative Selection

To effectively adapt the retriever to the new distribution, we desire to select support and diverse negatives for model training. Thus, we define *positive sample superiority* (*PSS*) and *inter sample diversity* (*ISD*) criteria to instruct the data selection in each step.

Let $\boldsymbol{q}$ and $\boldsymbol{d}$ denote the embedding of query q and document d, and $\boldsymbol{d}_{\parallel \boldsymbol{q}}$ and $\boldsymbol{d}_{\perp \boldsymbol{q}}$ denote the projection of $\boldsymbol{d}$ on the directions that are horizontal and vertical to $\boldsymbol{q}$. Intuitively, $\boldsymbol{d}_{\parallel \boldsymbol{q}}$ and $\boldsymbol{d}_{\perp \boldsymbol{q}}$ represent the information in d that is related and unrelated to q respectively.

Definition 1 (Positive Sample Superiority). The *positive sample superiority* between d and d_q^+ for the query q is given by

$$PSS(d, d_q^+; q) = sign(\boldsymbol{d}_{\parallel \boldsymbol{q}}^+ - \boldsymbol{d}_{\parallel \boldsymbol{q}}) \cdot \left\| \boldsymbol{d}_{\parallel \boldsymbol{q}}^+ - \boldsymbol{d}_{\parallel \boldsymbol{q}} \right\|_2, \tag{2}$$

where $sign(\cdot) = 1$ if $\boldsymbol{d}_{\parallel \boldsymbol{q}}^+ - \boldsymbol{d}_{\parallel \boldsymbol{q}}$ and $\boldsymbol{d}_{\parallel \boldsymbol{q}}^+$ are in the same direction and -1 otherwise, $\|\cdot\|_2$ is the ℓ_2 norm, d can be any document and d_q^+ is a relevant document for q, and $\boldsymbol{d}_{\parallel \boldsymbol{q}}$ is defined as

$$\boldsymbol{d}_{\parallel \boldsymbol{q}} = \frac{(\boldsymbol{d} \cdot \boldsymbol{q}) * \boldsymbol{q}}{|\boldsymbol{q}| * |\boldsymbol{q}|}. \tag{3}$$

The *PSS* measures the superiority of d_q^+ being more relevant to the query q than d, by comparing the differences between their information related to q. Therefore, a higher *PSS* value suggests that d is less likely to be an unlabeled relevant sample for q.

Definition 2 (Inter Sample Diversity). For a given query q, the *inter sample diversity* between d and a document set D is

$$ISD(d, D; q) = \frac{1}{|D|} \sum_{d' \in D} \left\| \boldsymbol{d}_{\perp \boldsymbol{q}} - \boldsymbol{d}'_{\perp \boldsymbol{q}} \right\|_2, \tag{4}$$

where $\boldsymbol{d}_{\perp \boldsymbol{q}} = \boldsymbol{d} - \boldsymbol{d}_{\parallel \boldsymbol{q}}$. The *ISD* measures the diversity of document d relative to the document set D, by comparing the differences between the information unrelated to q among the documents.

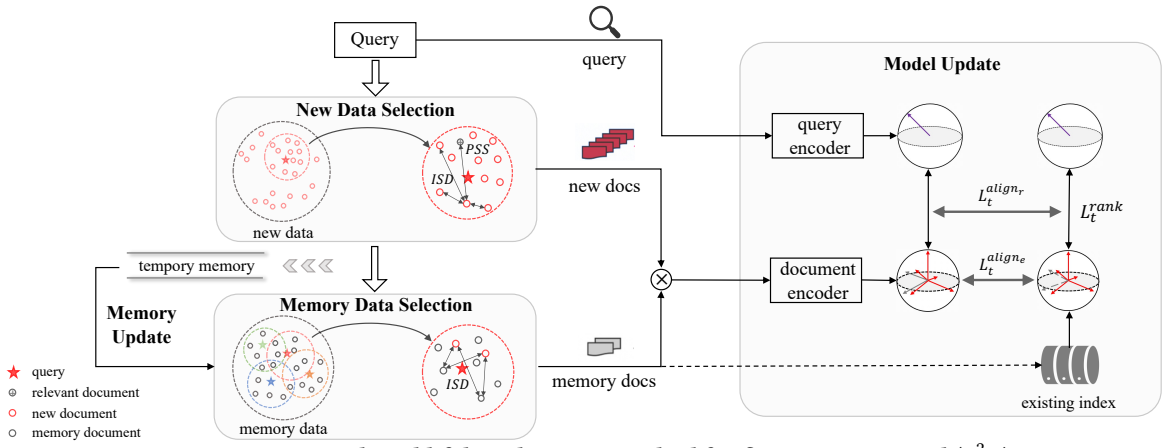

Figure 1: Memory-based lifelong learning method for first-stage retrieval (L^{2}R).

Based on the two defined criteria, we introduce each step of the model learning, taking $(q, d_q^+) \sim C_0$ in session t as an example.

STEP 1: New Data Selection. For using the new data to adapt to the current session t, we have the following principles: (1) Documents that are likely to be unlabeled positives should be avoided during selection, since mistakenly identifying relevant documents for model learning could cause serious damage to the performance. (2) The selected documents should be the negatives that can support the model to learn the decision boundary (we refer to them as support negatives). Such documents should be not trivial for the model to differentiate. (3) The selected documents should have minimum redundancy. With these principles, we propose the following selection strategy for the new data.

We first retrieve top results for q from the new-coming document collection $\mathcal{D}_t$ with BM25 to filter out massive non-informative samples, and obtain its potential support samples D_q^S. Then, based on the defined *PSS* and *ISD* criteria, we adaptively select n_1 diverse support negatives from D_q^S with:

$$D_q^{new} = \left\{ \arg\max_{d \in D_q^S}^{(n_1)} \ \alpha \cdot PSS(d, d_q^+; q) + (1-\alpha) \cdot ISD(d, D_q^S; q) \right\}, \quad (5)$$

where the embedding $\boldsymbol{q}$ and $\boldsymbol{d}$ used to calculate *PSS* and *ISD* are obtained from the latest model f_t. The *PSS* component helps to bypass unlabeled relevant documents and the *ISD* component prefers the samples that are distinct from the majority. We use a hyperparameter α to reconcile the two measures. With Eq. (5), we select n_1 new documents D_q^{new} from $\mathcal{D}_t$ that satisfy the aforementioned three principles. These selected samples are reserved for model updating and also added to the temporary memory M_c as candidates to update the memory M_{t-1}.

STEP 2: Memory Data Selection. To prevent the model from forgetting old knowledge when learning from the new data $\mathcal{D}_t$, we also select replay samples for model training from M_{t-1} that are: (1) pivotal for learning the historical versions of the model, i.e., historical support samples; (2) not redundant with each other for efficiency concerns; (3) different from the selected samples in D_q^{new} to better balance the acquired knowledge and new knowledge.

With the memory updating strategy in STEP 4, the samples in M_{t-1} already satisfy the first two desiderata. To filter with the third principle, we select n_2 documents D_q^{mem} from M_{t-1} that have the maximum *ISD* score regarding D_q^{new}:

$$D_q^{mem} = \left\{ \arg\max_{d \in D_q^O}^{(n_2)} \ ISD(d, D_q^{new}; q) \right\}, \quad (6)$$

where D_q^O denotes the stored old documents for q in the memory buffer M_{t-1}. Note that for computing the *ISD* score, the embedding $\boldsymbol{d}$ of memory samples are the existing ones computed in previous sessions when the learning is backward-compatible. Otherwise, the embedding is obtained using the latest model f_t. In the rest of this paper, we adopt the same approach to compute *ISD*, and we will omit this reminder unless there are special circumstances.

STEP 3: Model Update. With the selected new documents D_q^{new} (from STEP 1) and memory documents D_q^{mem} (from STEP 2), we can update a standard retrieval model from f_{t-1} to f_t. Without loss of generality, the retrieval model f_t can be formalized as,

$$f_t(q, d) = \langle \mathrm{E}_t^q(q), \mathrm{E}_t^d(d) \rangle, \quad (7)$$

where E_t^q and E_t^d are the query and document encoders, and the dot-product function is used to calculate the relevance score based on their embeddings. For model training, we use the standard contrastive learning objective [15, 45] to compute the loss for the positive document d_q^+ (no compatibility is ensured)[1]:

$$L_t^{no-com} = -\log \frac{\exp(f_t(q, d_q^+))}{\sum_{d \in \{d_q^+\} \cup D_q^{new} \cup D_q^{mem}} \exp(f_t(q, d))}. \quad (8)$$

When the model updating is not backward-compatible for document representations, we need to re-embed all the documents up to the t-th session, i.e., $D_{0:t}$, with f_t to rebuild the retrieval index. To eliminate the need for re-inferring embeddings of old documents, we can replace the learning objective in Eq. (8) with the backward-compatible learning objective in Section 4.3.

STEP 4: Memory Update. In practice, the memory buffer size is often limited to ensure efficiency in selecting replay samples, even though it does not impose a heavy storage burden. Given the limited budget n for each query, selecting which samples to include or replace in the memory is critical. We consider two principles to populate the memory: (1) The sample should have a strong impact on the learning of the decision boundary; (2) The redundancy between stored samples should be minimized. In contrast to most work that updates the memory in each training step [1, 5], we delay the memory update until after the completion of model updating in each session in order not to occupy the limited slots in the buffer.

[1]Here, we omit the in-batch negatives in Eq. (8) for brevity.

To preserve important information of the current session t for the future, we follow the first principle and consider only the support samples in M_c as candidates to update M_{t-1}. We calculate the *ISD* score of documents in M_{t-1} and M_c regarding k randomly-sampled anchor documents in M_{t-1}, and use the new documents with the maximum diversity to replace n_3 memory samples with the minimum diversity. Finally, we empty the temporary memory buffer to prepare for the next session. Note that, for the initial session ($t=0$), we use reservoir sampling [5] to fill the memory.

4.3 Backward-compatible Learning

To save the cost of repeated embedding computation, it is desirable for the model updating to ensure backward-compatibility of document representations. It means that existing embeddings for $\mathcal{D}_{0:t-1}$ do not need to be updated, and only the embeddings of new documents in $\mathcal{D}_t$ are computed with the lastest model f_t to update the index. We first introduce a vanilla method that can ensure backward compatibility, and two auxiliary alignment objectives for effective compatible learning.

Vanilla Compatible Learning. A straightforward approach is to optimize a new contrastive learning loss by fixing the embeddings of previous documents (i.e., the positive sample and the memory samples selected in the current training):

$$L_t^{rank} = -\log \frac{\exp(\langle \mathrm{E}_t^q(q), \boldsymbol{d}_q^+ \rangle)}{Z}, \tag{9}$$

where Z is a normalization term:

$$Z = \sum_{d \in \{d_q^+\} \cup D_q^{mem}} \exp(\langle \mathrm{E}_t^q(q), \boldsymbol{d} \rangle) + \sum_{d \in D_q^{new}} \exp(f_t(q,d)). \tag{10}$$

The Eq. (9) optimizes the model on the new data and existing document embeddings in a unified space to ensure compatibility. However, since all the new samples in D_q^{new} are negatives and only the embeddings of new samples are learnable, the model could easily learn the wrong correlation between a document being in the new distribution and it being irrelevant, leading to significant performance regression (see the experimental results in Section 6). In order to facilitate effective backward-compatible representation learning, we introduce two auxiliary alignment objectives.

Embedding-aligned Learning. As in [37], a common approach to ensure backward-compatible model updating is to minimize the ℓ_2 distance between the embeddings of previous documents (i.e., $\{d_q^+\} \cup D_q^{mem}$) calculated with the new model f_t and their existing embeddings:

$$L_t^{align_e} = \sum_{d \in \{d_q^+\} \cup D_q^{mem}} \frac{1}{2} \left\| \mathrm{E}_t^d(d) - \boldsymbol{d} \right\|_2^2. \tag{11}$$

By guiding the model to encode the old documents similarly to their existing embeddings, it could urge the model to learn decent document representations instead of blindly demoting new documents. However, this pointwise alignment is too strict for the model to adapt to the new documents sufficiently.

Ranking-aligned Learning. To relax the constraint on the model to learn new knowledge, we propose a loose listwise alignment objective. The goal is to minimize the divergence between the predicted distributions of the candidate documents calculated based on the existing and currently learned embeddings, i.e., $p(D \mid q)$ and $p'(D \mid q)$, respectively:

$$L_t^{align_r} = \mathrm{KL}\left(p(D \mid q) \,\|\, p'(D \mid q)\right) = \sum_{d \in D} p(d \mid q) \log \frac{p(d \mid q)}{p'(d \mid q)}, \tag{12}$$

where $D = \{d_q^+\} \cup D_q^{mem} \cup D_q^{new}$, and

$$p(d \mid q) = \begin{cases} \frac{\exp(f_t(q,d))}{Z} & \text{if } d \in D_q^{new} \\ \frac{\exp(\langle \mathrm{E}_t^q(q), \boldsymbol{d} \rangle)}{Z} & \text{if } d \in \{d_q^+\} \cup D_q^{mem} \end{cases}, \tag{13}$$

$$p'(d \mid q) = \frac{\exp(f_t(q,d))}{\sum_{d \in D} \exp(f_t(q,d))}. \tag{14}$$

The probability distribution $p(D \mid q)$ represents the model inference when backward compatibility is enabled, and $p'(D \mid q)$ represents the model predictions without compatible learning where all the embeddings need to be learned. In contrast to the pointwise embedding alignment, this ranking-based alignment not only allows more flexible exploration in the representation space but also facilitates bidirectional supervision for model learning: 1) $p'(D \mid q)$ can adapt the model better to the new data since the embeddings of candidates are all currently learned including the new ones. So it could guide $p(D \mid q)$, the backward-compatible inference we finally use, to better acquire new knowledge. 2) In $p(D \mid q)$, since the positive document and memory negative samples are ranked based on their existing embeddings up until session $t-1$, $p(D \mid q)$ captures their relative rankings from the model at session $t-1$. This older model has seen the negatives from session 1 to $t-1$ including the ones that have been removed from the memory, which could cover various types of negatives. Hence, by aligning with $p(D \mid q)$, $p'(D \mid q)$ can learn from the older more experienced model. Given the above mutual supervisions between the new and old model, our proposed ranking alignment objective ensures the representation compatibility without compromising the model performance, obtaining even better results (see more analysis in Section 6.3).

Overall Compatible Learning Objective. The final training objective to enable backward compatibility is the combination of the vanilla ranking loss and the alignment loss:

$$L_t^{com} = L_t^{rank} + \lambda \cdot L_t^{align}, \tag{15}$$

where L_t^{align} is either $L_t^{align_e}$ or $L_t^{align_r}$, and λ is a hyper-parameter to control the effect of the alignment regularization.

5 EXPERIMENTAL SETTINGS

5.1 Benchmark Construction

There are no publicly available datasets that could show the continuous growth of documents in realistic retrieval scenarios, potentially with distribution drift, booming events, and newly emerged relevant documents to previous queries. Thus, we build two benchmarks, i.e., LL-LoTTE and LL-MultiCPR, based on two retrieval datasets LoTTE [36] and Multi-CPR [22], to simulate the scenario with the aforementioned properties through the following steps:

Preprocessing. LoTTE and Multi-CPR are two retrieval datasets that consist of 5 and 3 domains with separate subsets of documents and queries respectively. For each domain of the two datasets, we merge all the data and re-split them into train/dev/test sets with a ratio of 0.7:0.15:0.15 for LoTTE and 0.9:0.05:0.05 for Multi-CPR. Table 1 lists the statistics of the two datasets.

Session Partitioning. We build an initial collection $\mathcal{D}_0$ and 3 upcoming sessions with different document distributions for both LL-LoTTE and LL-MultiPCR. In LL-LoTTE, we use technology and writing as the common domains where documents emerge evenly over time, and lifestyle, recreation, and science as the booming domains in each of the upcoming sessions. We keep 70% and 40% of

Table 1: Statistical information of LoTTE and Multi-CPR.

domain	#query	#document	len_q	len_d	#qrels
LoTTE					
technology	5519	1,914,731	9.00	124.69	6.59
writing	5571	477,066	9.11	171.39	5.89
lifestyle	5156	388,354	10.04	166.65	5.10
recreation	5491	430,000	9.16	193.16	4.27
science	5185	2,037,806	9.08	139.83	5.98
Multi-CPR					
e-commerce	101,000	1,002,822	6.90	32.96	1.0
medical	100,999	959,526	17.07	121.90	1.0
entertainment	101,000	1,000,000	7.41	27.45	1.0

the random documents from the common and booming domains respectively in $\mathcal{D}_0$. Next, we construct 3 corpora $\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}$ for the following three sessions. Each corpus consists of 10%, 50%, and 5% of documents from the two common domains, a booming domain, and the remaining two domains respectively. With the documents in each session, we collect their connected queries from the new split train/dev/test sets of LoTTE to construct the training dataset and dev/test sets. Note that, under the setting where new-coming documents have no labels, the labeled relevant query-document pairs for model training remains the C_0, but the dev/test sets Q_t^{dev} and $Q_t^{\text{test}} (\forall t \in \{0, \cdots, 3\})$ can have more annotated relevant documents for evaluation. In LL-MultiCPR, similar to LL-LoTTE, we choose e-commerce as the common domain, medical and entertainment as the booming domains for Session 1 and 2 respectively. Since there are only three domains, Session 3 has no booming domains and simply includes the remaining documents.

Postprocessing. In LoTTE, almost all relevant documents of each query have positive labels. This makes it hard to simulate the realistic scenario where quite a few relevant documents to training queries may appear in the upcoming sessions and not be labeled. To overcome this issue, we collect extra pseudo-relevant documents for training queries using OpenAI API (text-davinci-003), and distribute these unlabeled documents to each coming session with the same sampling ratios in session partitioning. Specifically, we use two types of instructions for pseudo-relevant document generation: (1) "Given a question $\{q\}$ and a relevant document $\{d_q^+\}$, please generate 5 other relevant documents."; (2) "Given a document $\{d_q^+\}$, please rephrase it.". Through this process, we obtain approximately 18.5 documents for each training query[2]. For LL-MultiCPR, we have not conducted post-processing since there are sizable unlabeled relevant documents in Multi-CPR (see [22]).

Table 2 lists the statistics of the final LL-LoTTE and LL-MultiCPR datasets. Following similar steps, other existing retrieval datasets can also be transformed to evaluate lifelong learning of first-stage retrieval. When there are no explicitly separate domains, topic clustering could be applied for simulation and we leave such investigation for future research.

5.2 Evaluation Metrics

We define metrics to evaluate lifelong learning methods for first-stage retrieval. Considering the realistic scenario, for each session, we care more about the retrieval performance on the queries in the current session. Let $p_{i,j}$ be the retrieval performance evaluated on the test queries of session j (i.e, Q_j^{test}) over the document collection $\mathcal{D}_{0:j}$ after the learning of session i, and $p_{i,j}$ can be measured using any common retrieval metric like Recall or MRR. We take the performance at session t, namely P_t, and average performance over all coming sessions, namely AP, to compare various methods:

$$\mathrm{P_t} = p_{t,t}, \qquad \mathrm{AP} = \frac{1}{T}\sum_{t=1}^{T} p_{t,t}. \tag{16}$$

Following [25], we also apply auxiliary metrics to assess how fast a model learns (Training Time), how much the model forgets ($\mathrm{Forget_t}$), and how well the model transfers knowledge from one session to future sessions (FWT). Formally, they are defined as:

$$\mathrm{Forget_t} = \frac{1}{t}\sum_{j=0}^{t-1} \max_{l\in\{0,\ldots,t-1\}} \left(p_{l,j} - p_{t,j}\right), \quad \mathrm{FWT} = \frac{\sum_{i=1}^{j-1}\sum_{j=2}^{T} p_{i,j}}{\frac{T(T-1)}{2}}. \tag{17}$$

To instantiate the above metrics in our work, we consider the evaluation method of the original LoTTE [36] and Multi-CPR [22]. Besides Recall (R@N), Success (S@N) and Mean Reciprocal Rank (MRR@N) are used in LoTTE and Multi-CPR respectively. Following the official cutoffs for N, we show the lifelong learning performance on the above defined metrics regarding S@5 and R@100 for LL-LoTTE, and MRR@10 and R@1000 for LL-MultiCPR.

Table 2: Statistics of LL-LoTTE and LL-MultiCPR datasets.

	$\mathrm{Session_0}$	$\mathrm{Session_1}$	$\mathrm{Session_2}$	$\mathrm{Session_3}$
LL-LoTTE				
#document	2,816,720	654,266	670,026	1,405,225
#train_q	16,147	-	-	-
#dev_q	3449	1681	1750	1666
#test_q	3448	1707	1752	1700
#test_qrels	4.16	6.79	7.55	8.31
LL-MultiCPR				
#document	1,486,184	630,545	648,307	198,310
#train_q	136,282	-	-	-
#dev_q	7551	3340	3270	989
#test_q	7653	3242	3223	1032
#test_qrels	1.0	1.0	1.0	1.0

5.3 Baselines

We consider two types of baselines for comparison:

Memory-based Methods: (1) ER [5] applies random sampling for memory data selection and reservoir sampling for memory update. Despite its simplicity, ER outperforms many complex lifelong learning methods [25]. (2) MIR [1] chooses replay samples according to their loss increment regarding the updated model learned on the new data, and also uses reservoir sampling for memory update. (3) GSS [2] has the same memory data selection strategy as ER but refines the memory update by trying to diversify the samples in the memory buffer based on their gradients. However, it incurs huge computation costs. (4) OCS [46] is one of the latest methods for lifelong learning containing noisy data. It selects high-affinity samples to previous data based on their gradients for model and memory update.

Naive Methods: (1) *Initial* conducts no model updating and uses the model trained in the initial session for the retrieval in the upcoming sessions. (2) *Incre-train* initializes the model training from the previous session and updates it with the new data in the current session. (3) *Retrain* trains the model from scratch in each session using the whole available data until that session.

[2]We perform quantitative analysis on these generated pseudo documents to ensure that they are of high quality. It shows that 63% of them can be retrieved in the top-200 results of BM25 for training queries in each upcoming session.

To see the separate effect of our proposed data selection strategy and ranking alignment objective, we compare our method with the baselines both without and with backward-compatible representation learning (based on Eq. (8) or Eq. (15)). Note that: (1) *Initial* has the same performance under the two settings since the model is not updated; (2) *Retrain* works only without backward compatibility since the model is retrained from scratch in each session. For the comparisons with the backward-compatibility constraint, we equip the baselines with the embedding alignment objective in Eq. (15), and our model L^{2}R uses no or one of the two alignment objectives, named as $\mathbf{L^2R}_{vanilla}$, $\mathbf{L^2R}_{emb}$, and $\mathbf{L^2R}_{rank}$, respectively.

5.4 Implementation Details

We implement the retrieval model with DPR [15], and the parameters are initialized with BERT-base released by Google. The hyper-parameters in baselines and our method are tuned on the dev set. For LL-LoTTE, we truncate the input query and passage to a maximum of 32 and 256 tokens respectively. We train retrieval models with BM25 top-500 results for the initial session and top-200 results for the upcoming sessions, and the key hyper-parameters of BM25 are tuned to $k_1 = 0.80$ and $b = 0.72$. We use a batch size of 96, and a learning rate of 5e-6 and 1e-6 for the initial session and upcoming sessions respectively. For LL-MultiCPR, we set the query and passage length to 32 and 128 respectively. We train the initial session and upcoming sessions with BM25 top-500 results, with $k_1 = 0.20$ and $b = 0.72$. We use a learning rate of 1e-5 and 3e-6 for the initial and upcoming sessions respectively, and a batch size of 192. For the two datasets, we pair each positive document with 5 negatives for training, including 3 new documents and 2 memory documents. For data selection, we upsample a subset with twice the desired number of documents in each training step, instead of the entire collection, to save the computation cost. For memory update, we set the number of anchor documents and replaced documents are 1/3 of the memory buffer size n. We set α to 0.6 and 0.8 for LL-LoTTE and LL-MultiCPR respectively, and λ to 1.0 and 3.0 for both. For each dataset, we set the buffer size n of each training query with two settings that can hold: (1) half of training negatives in the initial session (i.e., 30 for LL-LoTTE and 10 for LL-MultiCPR); (2) total training negatives in all the sessions (i.e., 100 for LL-LoTTE and 30 for LL-MultiCPR). We use the former as the default setting.

We adopt the Transformers for implementations and all experiments run on Nvidia Tesla V100-32GB GPUs. Statistically significant differences are measured by a two-tailed t-test. The datasets and code are available at https://github.com/caiyinqiong/L-2R.

6 RESULTS AND DISCUSSION

In this section, we present the experimental results and conduct thorough analysis of L^{2}R to clarify its advantages.

6.1 Main Evaluation

We compare the performance of L^{2}R with all the baselines in Section 5.3, and record their results under both settings in Table 3 & 4.

Performance without Representation Compatibility. From Table 3, we find that: (1) Without special measures for lifelong learning, neural retriever DPR (i.e., *Initial*) shows poorer generalization ability than the term-based retrievers (i.e., BM25), especially when the distribution changes violently. For example, DPR outperforms BM25 on LL-LoTTE in Session 0-2 but underperforms it when massive science documents influx in Session 3 (note that there are significantly more documents in the science domain than others). This observation is consistent with the conclusion in [33] that neural retrievers are less robust than BM25. (2) For the methods that learn from new data (i.e., methods except *Initial*), *Incretrain* performs poorly than memory-based methods, probably because it does nothing to address the catastrophic forgetting issue. Additionally, *Incre-train* is not always superior to *Initial*, particularly on LL-MultiCPR. Apart from the forgetting issue, we believe a potential reason is that the sizable unlabeled relevant documents in the new data could hurt model updating. (3) It is worth noting that *Retrain* exhibits worse performance, particularly on recall, which deviates from findings in other lifelong learning tasks like image classification [1]. It is probably because the retrained retriever has seen fewer varieties of negative samples and has a higher probability of using emerged unlabeled positive documents for training. (4) Among the memory-based methods, MIR has the overall best performance on both datasets and OCS can not exceed it. This shows that the gradient-based method to filter out noisy data in OCS does not work effectively for unlabeled relevant documents in the retrieval task. (5) On both benchmarks, L^{2}R consistently outperforms the baselines in all upcoming sessions. Especially in Session 3 of LL-LoTTE that has violent distribution drift, L^{2}R beats others by a large margin and surpasses BM25. These performance gains confirm the advantages of our proposed data selection strategy in L^{2}R.

Performance with Representation Compatibility. The performance of all the methods with representation compatibility is presented in Table 4. Compared to Table 3, we have the following observations. From the perspective of **effectiveness**: (1) Adding the embedding alignment to ensure representation compatibility leads to significantly lower model performance, even worse than *Initial*. It shows that enforced embedding alignment could hurt model learning on new data. Among these methods, *Incre-train* is hurt the least, probably because the regularization is applied on fewer documents. (2) For the three variants of L^{2}R with representation compatibility, L^{2}R$_{vanilla}$ suffers from model collapse by only optimized with the contrastive learning loss on existing embeddings of previous documents. By injecting an alignment regularization, the model could be updated more effectively with backward-compatible representations. (3) It is exciting that L^{2}R$_{rank}$ can significantly exceed L^{2}R$_{emb}$, and even outperform L^{2}R that without representation compatibility in some sessions (e.g., Session 1 of LL-LoTTE and almost all the sessions of LL-MultiCPR). It shows that the alignment on predicted ranking lists allows for more flexible encoder updates than direct embedding alignment, and the prediction results based on existing embeddings (computed by old models) provide beneficial information based on the previously acquired knowledge to guide the model to learn new data (see further analysis in Section 6.3). From the perspective of **efficiency**: (1) With representation backward-compatibility, it can save 79% (2.73M *vs.* 13.16M) and 81% (1.47M *vs.* 7.85M) of computation costs for inferring document representations than that without compatibility on LL-LoTTE and LL-MultiCPR respectively (accumulated on 3 upcoming sessions). Overall, these results demonstrate that the ranking alignment objective in L^{2}R could promote both the effectiveness and efficiency of model lifelong learning.

Table 3: Evaluation results on LL-LoTTE and LL-MultiCPR without representation compatibility. Bold and underline indicate the best overall and baseline performance. $*$ indicates statistically significant improvements over all baselines ($p < 0.05$).

Method	LL-LoTTE										LL-MultiCPR									
	S@5					R@100					MRR@10					R@1000				
	P_0	P_1	P_2	P_3	AP	P_0	P_1	P_2	P_3	AP	P_0	P_1	P_2	P_3	AP	P_0	P_1	P_2	P_3	AP
BM25	40.0	45.3	43.6	44.5	44.5	47.1	43.0	40.2	37.6	40.3	19.69	14.78	17.04	15.97	15.93	72.43	51.79	73.35	68.99	64.71
Initial	41.0	47.4	44.3	41.6	44.4	48.4	43.7	41.7	35.4	40.3	25.16	16.22	20.79	19.88	18.96	83.88	66.66	79.77	78.20	74.88
Incre-train	/	47.3	45.5	42.1	45.0	/	43.7	41.9	35.1	40.2	/	15.32	20.67	19.85	18.61	/	65.21	78.68	78.07	73.99
Retrain	/	47.4	44.5	41.0	44.3	/	43.6	40.2	33.9	39.2	/	15.51	20.25	19.50	18.42	/	64.37	78.93	76.65	73.32
ER	/	47.8	45.4	42.6	45.3	/	44.0	42.0	35.3	40.4	/	16.15	20.87	20.14	19.05	/	66.93	79.74	78.59	75.09
MIR	/	48.7	46.1	43.4	46.1	/	44.2	42.7	36.0	41.0	/	16.07	21.01	20.32	19.13	/	66.90	79.77	78.49	75.05
GSS	/	48.3	45.8	43.3	45.8	/	44.2	42.3	35.4	40.6	/	16.43	20.78	19.95	19.05	/	67.02	79.37	78.71	75.03
OCS	/	48.6	46.1	43.4	46.0	/	44.3	42.5	35.9	40.9	/	16.39	20.57	20.22	19.06	/	66.75	79.46	78.29	74.83
L^2R	/	**50.0***	**48.0***	**46.5***	**48.2***	/	**45.9***	**44.5***	**38.2***	**42.9***	/	**17.25***	**22.34***	**21.57***	**20.39***	/	**68.69***	**80.55***	**80.17***	**76.47***

Table 4: Evaluation results on LL-LoTTE and LL-MultiCPR with representation compatibility. Bold and underline indicate the best overall and baseline performance. $*$ indicates statistically significant improvements over all baselines ($p < 0.05$).

Method	LL-LoTTE								LL-MultiCPR							
	S@5				R@100				MRR@10				R@1000			
	P_1	P_2	P_3	AP	P_1	P_2	P_3	AP	P_1	P_2	P_3	AP	P_1	P_2	P_3	AP
Initial	47.4	44.3	41.6	44.4	43.7	41.7	35.4	40.3	16.22	20.79	19.88	18.96	66.66	79.77	78.20	74.88
Incre-train	44.9	42.2	39.0	42.0	41.1	39.1	33.9	38.0	8.85	14.54	12.11	11.83	57.77	76.02	72.45	68.75
ER	45.2	41.6	37.8	41.5	41.4	38.3	32.8	37.5	9.31	13.61	11.66	11.53	57.80	75.18	71.03	68.00
MIR	45.2	42.2	38.3	41.9	41.5	38.5	33.0	37.7	9.23	13.62	11.50	11.45	57.87	75.09	70.93	67.96
GSS	45.1	42.1	38.2	41.8	41.4	38.8	33.1	37.8	9.29	13.46	11.32	11.36	57.95	75.37	71.14	68.15
OCS	45.1	42.3	38.3	41.9	41.5	38.8	33.2	37.8	9.28	13.16	11.42	11.29	57.77	75.01	70.89	67.89
$L^2R_{vanilla}$	40.3	36.8	33.3	36.8	31.0	27.0	22.0	26.7	3.31	9.64	7.75	6.90	23.87	47.16	41.67	37.57
L^2R_{emb}	46.3	43.8	38.8	43.0	42.4	38.7	33.4	38.2	9.38	14.20	12.27	11.95	58.11	75.80	72.38	68.76
L^2R_{rank}	**50.6***	**47.3***	**44.6***	**47.5***	**46.9***	**44.1***	**37.8***	**42.9***	**22.61***	**25.80***	**29.11***	**25.84***	**70.64***	**80.05***	**80.91***	**77.20***

Table 5: Evaluation results of the last session (P_3) with different buffer size on LL-LoTTE and LL-MultiCPR. All the methods run with representation compatibility.

Method	LL-LoTTE				LL-MultiCPR			
	n=30		n=100		n=10		n=30	
	S@5	R@100	S@5	R@100	MRR@10	R@1000	MRR@10	R@1000
ER	37.8	32.8	38.1	32.9	11.66	71.03	11.71	71.32
MIR	38.3	33.0	37.9	33.0	11.50	70.93	11.72	71.41
GSS	38.2	33.1	38.4	33.1	11.32	71.14	11.44	71.33
OCS	38.3	33.2	38.8	33.1	11.42	70.89	11.43	71.01
L^2R_{rank}	44.6	37.8	45.2	38.2	29.11	80.91	30.27	81.30

Performance with Larger Memory Buffer Size. To investigate the impact of memory buffer size on model performance, we conduct experiments using a larger buffer that can hold all the training samples used in the four sessions. Only the results of the last session using different n are reported in Table 5 for a clear comparison. We observe that with a larger buffer size, the performance of L^2R_{rank} is further improved, particularly on precision metrics such as S@5 and MRR@10 (e.g., the improvement is 1.3% on S@5 for LL-LoTTE and 4.0% on MRR@10 for LL-MultiCPR.). However, the baseline methods do not benefit as much from a larger memory, probably because ER and MIR store random-sampled documents, and more importantly, all of them cannot filter out the unlabeled positives. In contrast, L^2R stores diverse support negative samples, thereby making more efficient use of the memory buffer slots.

6.2 Studies on Data Selection Strategy

We run ablation studies on the data selection strategy to investigate its impact on model learning.

For data selection, we define *PSS* and *ISD* to measure the likelihood of a document being negative and its diversity relative to others. We compare L^2R_{rank} with several ablation variants to verify the effectiveness of our criteria in Table 6: (1) For the *NewDataSelection* module, we observe that without the *PSS* component to filter out unlabeled relevant documents in the new data, the retrieval performance on the two datasets significantly decreases. Removing *ISD* also causes a performance drop, especially on recall, since the retriever has seen fewer varieties of negative samples if redundancy among the selected samples is not considered. These results demonstrate that both criteria are important in selecting new data for the model to adapt to new distributions. (2) For the *MemoryDataSelection* module, we remove the *ISD* component and randomly select replay samples from the memory. The performance decreases on both datasets, showing that selecting samples different from the new data for replaying is critical for effective model updating. It is probably because the cooccurrence of discrepant or even conflicting data encourages the model to deliberate the balance between learning new knowledge and preserving old knowledge. (3) For the *MemoryUpdate* module, we remove the *ISD* component and replace the samples in the memory randomly. The results show that LL-LoTTE has less regression in performance compared to LL-MultiCPR, probably because LL-MultiCPR uses a

Table 6: Ablation studies on the data selection strategy in L^{2}R$_{rank}$. Evaluation results of the last session (P_3) in LL-LoTTE and LL-MultiCPR are reported.

Module	Strategy	LL-LoTTE		LL-MultiCPR	
		S@5	R@100	MRR@10	R@1000
L^2R_{rank}		44.6	37.8	29.11	80.91
NewDataSelection	*-PSS*	43.8	37.3	28.99	79.65
	-ISD	44.3	37.4	29.14	80.62
	-Both	43.6	37.2	28.82	79.36
MemoryDataSelection	*-ISD*	44.4	37.1	28.19	79.75
MemoryUpdate	*-ISD*	44.3	37.5	28.59	80.10

Table 7: Investigations on the alignment objectives. Evaluation of each session ($P_1 - P_3$) in LL-LoTTE are reported.

Method	Query	L^{2}R		L^{2}R$_{emb}$		L^{2}R$_{rank}$	
		S@5	R@100	S@5	R@100	S@5	R@100
Session$_1$	#seen: 1469	51.1	43.3	48.5	41.5	50.9	44.2
	#unseen: 238	40.3	57.1	32.4	47.5	48.7	63.3
Session$_2$	#seen: 1525	48.9	41.3	46.6	37.4	47.8	41.6
	#unseen: 227	39.2	58.0	25.1	47.3	44.1	60.6
Session$_3$	#seen: 1573	47.7	37.5	40.6	33.0	45.3	36.9
	#unseen: 127	30.7	47.6	17.3	38.1	36.2	50.2

smaller buffer size (n=10), and storing non-redundant samples becomes more important for it to address the forgetting issue.

6.3 Studies on Alignment Objectives

We conduct studies on the embedding and ranking alignment objectives to probe their impact on model updating.

Performance on Seen and Unseen Queries. To understand how the alignment objectives affect model updating, we split the test set of each coming session in LL-LoTTE to previously seen queries and newly unseen queries, and evaluate the performance of L^{2}R, L^{2}R$_{emb}$, and L^{2}R$_{rank}$. From Table 7, we find that: (1) The seen queries generally achieve higher S@5 but lower R@100. It is because the seen queries usually have more relevant documents than the unseen queries, which is less favourable for them on recall. (2) In L^{2}R$_{emb}$, both seen and unseen queries experience a significant performance drop compared to L^{2}R that without compatibility. Especially, the drop on unseen queries is more dramatic than that on seen queries. It shows that direct embedding alignment constrains the model to learn new knowledge. (3) It is interesting that L^{2}R$_{rank}$ demonstrates improved performance on unseen queries compared to L^{2}R. It shows that the ranking results predicted on the old embeddings provide beneficial supervision to the model to learn relevance matching on new data. Moreover, the ranking alignment does not harm seen queries, unless the distribution changes drastically and the model compromises to fit new data (i.e., Session 3).

Performance on Auxiliary Metrics. We compare all the methods with representation compatibility on auxiliary metrics, including Forget$_T$, FWT, and Training Time, to gain insights into the model updating process. From Figure 2, we observe: (1) Among all the methods, L^{2}R$_{rank}$ performs best in addressing the catastrophic forgetting issue. Particularly on LL-LoTTE, it has negative values on Forget$_T$. Apart from the superior memory mechanism in L^{2}R, one possible reason is that, during the lifelong learning process, models with ranking-aligned compatible leaning could effectively acquire new knowledge, and the query encoder is adjusted

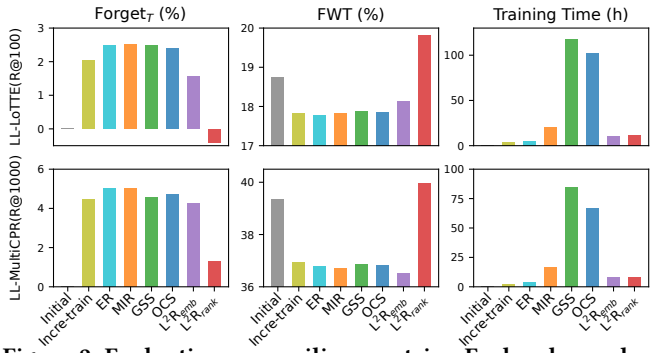

Figure 2: Evaluation on auxiliary metrics. Each column denotes a metric and each row denotes a dataset.

to better differentiate the relevant and irrelevant documents for test queries in historical sessions. (2) Besides the forgetting issue, L^{2}R$_{rank}$ shows promising forward transfer ability than the models optimized with embedding alignment, probably because tight embedding alignment with existing embeddings hinders model updating and generalizing to new queries. (3) On the training time, our methods have significantly lower training costs compared to GSS and OCS which require gradient calculations for each training sample and MIR which requires extra estimated model updating.

7 CONCLUSION AND FUTURE WORK

In this work, we study a common scenario in real-world search engines, where numerous documents are continuously emerging with potential distribution drift. To adapt the retriever to new distributions, we propose a memory-based lifelong learning method for first-stage retrieval (i.e., L^{2}R). By employing the selection strategy of *diverse support negatives* for model updating, along with a *ranking alignment objective* for backward-compatible representation learning, L^{2}R could continuously learn the retriever on unlabeled emerging documents both effectively and efficiently. Extensive experiments on our constructed benchmarks demonstrate the superiority of L^{2}R over competitive lifelong learning baselines.

Our work presents an initial step towards solving the critical challenges in lifelong learning for first-stage retrieval. Due to page limitations, certain promising directions remain unexplored in this study. Firstly, it is worth investigating whether the methods proposed for domain adaptation still work well in the lifelong learning setting, as both address the distribution changes. Secondly, the current method does not yet have specialized techniques to handle queries related to booming topics, which presents an avenue for future research. In conclusion, we believe that our study, despite its limited scope, provides valuable and generalizable insights that could guide future research on this task.

ACKNOWLEDGMENTS

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 61902381, the Youth Innovation Promotion Association CAS under Grants No. 2021100, the project under Grants No. JCKY2022130C039 and 2021QY1701, the CAS Project for Young Scientists in Basic Research under Grant No. YSBR-034, the Innovation Projectof ICT CAS under Grants No. E261090, and the Lenovo-CAS Joint Lab Youth Scientist Project.

REFERENCES

[1] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. 2019. Online Continual Learning with Maximal Interfered Retrieval. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 11849–11860.

[2] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. 2019. Gradient based sample selection for online continual learning. *Advances in neural information processing systems* 32 (2019).

[3] Magdalena Biesialska, Katarzyna Biesialska, and Marta R Costa-Jussa. 2020. Continual lifelong learning in natural language processing: A survey. *arXiv preprint arXiv:2012.09823* (2020).

[4] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of ECCV*. 532–547.

[5] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. 2019. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486* (2019).

[6] Zhiyuan Chen and Bing Liu. 2018. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 12, 3 (2018), 1–207.

[7] Mark Collier, Efi Kokiopoulou, Andrea Gesmundo, and Jesse Berent. 2020. Routing networks with co-training for continual learning. *arXiv preprint arXiv:2009.04381* (2020).

[8] Rahul Duggal, Hao Zhou, Shuo Yang, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. 2021. Compatibility-aware heterogeneous visual search. In *Proceedings of the IEEE/CVF conference on CVPR*. 10723–10732.

[9] Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, et al. 2022. Pre-training methods in information retrieval. *Foundations and Trends® in Information Retrieval* 16, 3 (2022), 178–317.

[10] Thomas Gerald and Laure Soulier. 2022. Continual Learning of Long Topic Sequences in Neural Information Retrieval. In *ECIR*. Springer, 244–259.

[11] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211* (2013).

[12] Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)* 40, 4 (2022), 1–42.

[13] Weihua Hu, Rajas Bansal, Kaidi Cao, Nikhil Rao, Karthik Subbian, and Jure Leskovec. 2022. Learning backward compatible embeddings. In *Proceedings of the 28th ACM SIGKDD*. 3018–3028.

[14] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[15] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[16] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.

[17] Jeremias Knoblauch, Hisham Husain, and Tom Diethe. 2020. Optimal continual learning has perfect memory and is np-hard. In *International Conference on Machine Learning*. PMLR, 5327–5337.

[18] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence* 40, 12 (2017), 2935–2947.

[19] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies* 14, 4 (2021), 1–325.

[20] Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Wei Chen, and Xueqi Cheng. 2023. On the Robustness of Generative Retrieval Models: An Out-of-Distribution Perspective. *arXiv preprint arXiv:2306.12756* (2023).

[21] Zheng Liu and Yingxia Shao. 2022. Retromae: Pre-training retrieval-oriented transformers via masked auto-encoder. *arXiv preprint arXiv:2205.12035* (2022).

[22] Dingkun Long, Qiong Gao, Kuan Zou, Guangwei Xu, Pengjun Xie, Ruijie Guo, Jian Xu, Guanjun Jiang, Luxi Xing, and Ping Yang. 2022. Multi-CPR: A Multi Domain Chinese Dataset for Passage Retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3046–3056.

[23] Jesús Lovón-Melgarejo, Laure Soulier, Karen Pinel-Sauvagnat, and Lynda Tamine. 2021. Studying catastrophic forgetting in neural ranking models. In *European Conference on Information Retrieval*. Springer, 375–390.

[24] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, and Xueqi Cheng. 2022. Pre-Train a Discriminative Text Encoder for Dense Retrieval via Contrastive Span Prediction. In *Proceedings of the 45th International ACM SIGIR* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 848–858.

[25] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. 2022. Online continual learning in image classification: An empirical survey. *Neurocomputing* 469 (2022), 28–51.

[26] Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Vol. 24. Elsevier, 109–165.

[27] Sanket Vaibhav Mehta, Jai Gupta, Yi Tay, Mostafa Dehghani, Vinh Q Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler. 2022. DSI++: Updating Transformer Memory with New Documents. *arXiv preprint arXiv:2212.09744* (2022).

[28] Qiang Meng, Chixiang Zhang, Xiaoqiang Xu, and Feng Zhou. 2021. Learning compatible embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9939–9948.

[29] Yujia Qin, Jiajie Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. Elle: Efficient lifelong pre-training for emerging data. *arXiv preprint arXiv:2203.06311* (2022).

[30] Kandan Ramakrishnan, Rameswar Panda, Quanfu Fan, John Henning, Aude Oliva, and Rogerio Feris. 2020. Relationship matters: Relation guided knowledge transfer for incremental learning of object detectors. In *Proceedings of the IEEE/CVF conference on CVPR workshops*. 250–251.

[31] Vivek Ramanujan, Pavan Kumar Anasosalu Vasu, Ali Farhadi, Oncel Tuzel, and Hadi Pouransari. 2022. Forward compatible training for large-scale embedding retrieval systems. In *Proceedings of the IEEE/CVF Conference on CVPR*. 19386–19395.

[32] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on CVPR*. 2001–2010.

[33] Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qifei Wu, Yuchen Ding, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2022. A thorough examination on zero-shot dense retrieval. *arXiv preprint arXiv:2204.12755* (2022).

[34] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[35] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671* (2016).

[36] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488* (2021).

[37] Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. 2020. Towards backward-compatible representation learning. In *Proceedings of the IEEE/CVF Conference on CVPR*. 6368–6377.

[38] Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. 2021. Online class-incremental continual learning with adversarial shapley value. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 9630–9638.

[39] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. 2017. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE international conference on computer vision*. 3400–3409.

[40] Yi Tay, Vinh Q Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. 2022. Transformer memory as a differentiable search index. *arXiv preprint arXiv:2202.06991* (2022).

[41] Gido M Van de Ven and Andreas S Tolias. 2019. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734* (2019).

[42] Timmy ST Wan, Jun-Cheng Chen, Tzer-Yi Wu, and Chu-Song Chen. 2022. Continual Learning for Visual Search with Backward Consistent Feature Embedding. In *Proceedings of the IEEE/CVF Conference on CVPR*. 16702–16711.

[43] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2023. A Comprehensive Survey of Continual Learning: Theory, Method and Application. *arXiv preprint arXiv:2302.00487* (2023).

[44] Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan-Fang Li, Guilin Qi, and Gholamreza Haffari. 2021. Pretrained language model in continual learning: A comparative study. In *International Conference on Learning Representations*.

[45] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[46] Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang. 2022. Online Coreset Selection for Rehearsal-based Continual Learning. In *The Tenth ICLR, 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

[47] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense text retrieval based on pretrained language models: A survey. *arXiv preprint arXiv:2211.14876* (2022).

[48] Kun Zhou, Xiao Liu, Yeyun Gong, Wayne Xin Zhao, Daxin Jiang, Nan Duan, and Ji-Rong Wen. 2022. MASTER: Multi-task Pre-trained Bottlenecked Masked Autoencoders are Better Dense Retrievers. *arXiv preprint arXiv:2212.07841* (2022).