DR.EHR: Dense Retrieval for Electronic Health Record with Knowledge Injection and Synthetic Data

Anonymous ACL submission

Abstract

Electronic Health Records (EHRs) are pivotal in clinical practices, yet their retrieval remains a challenge due to the reliance on exact match methods that fail to address semantic gaps. Recent advancements in dense retrieval offer promising solutions but existing models, both general-domain and biomedical-domain, fall short due to insufficient medical knowledge or mismatched training corpora. This paper introduces `DR.EHR`, a series of dense retrieval models specifically tailored for EHR retrieval. We propose a two-stage training pipeline utilizing MIMIC-IV discharge summaries to address the need for extensive medical knowledge and large-scale training data. The first stage involves medical entity extraction and knowledge injection from a biomedical knowledge graph, while the second stage employs large language models to generate diverse training data. We train two variants of `DR.EHR`, with 110M and 7B parameters, respectively. Evaluated on the CliniQ benchmark, our models significantly outperforms all existing dense retrievers, achieving state-of-the-art results. Detailed analyses confirm our models' superiority across various match and query types, particularly in challenging semantic matches like implication and abbreviation. Ablation studies validate the effectiveness of each pipeline component, underscoring the model's enhanced medical knowledge and adaptability to the EHR retrieval task. This work significantly advances EHR retrieval, offering a robust solution for clinical applications.

1 Introduction

Electronic Health Records (EHRs) hold significant value in various clinical practices, and EHR retrieval plays an crucial role in enabling physicians to utilize EHRs more efficiently (Zhang et al., 2019; Ying et al., 2025). This step is essential in a wide range of clinical tasks, including patient cohort selection (Jin et al., 2021; Yang et al., 2021), EHR Question Answering (QA) (Pampari et al., 2018; Lanz and Pecina, 2024), and patient chart review (Gupta et al., 2024; Ye et al., 2021).

Despite the critical importance of this field, its development has not progressed at a commensurate pace. Most existing EHR retrieval systems, whether in academic research or deployed in real-world hospitals, still rely on exact match methods (Ruppel et al., 2020; Negro-Calduch et al., 2021), which inevitably suffer from the semantic gap issue (Koopman et al., 2016; Edinger et al., 2012). A recent EHR retrieval benchmark, CliniQ (Zhao et al., 2025), which separately evaluates various matching types, quantitatively demonstrates that exact match methods struggle with semantic matches, even when augmented by query expansion using a Knowledge Graph (KG).

Recently, Dense Retrieval (DR), which leverages Pre-trained Language Models (PLMs) to generate dense text representations for retrieval, has garnered increasing research interest (Karpukhin et al., 2020). Owing to its inherent ability to capture semantics and large-scale contrastive learning, DR models have the potential to bridge the semantic gap and have exhibited strong zero-shot capabilities (Neelakantan et al., 2022; Xiao et al., 2023). In the context of EHR retrieval, general-domain models such as `bge` (Xiao et al., 2023) and `NV-Embed` (Lee et al., 2024) serve as strong baselines (Myers et al., 2024), but they leave significant room for improvement due to insufficient medical knowledge (Zhao et al., 2025). Biomedical-domain models, including `MedCPT` (Jin et al., 2023) and `BMRetriever` (Xu et al., 2024), also perform suboptimally despite ample knowledge, likely due to the mismatch between their training corpora and clinical notes. Thus, there is a pressing need for an EHR dense retriever specifically designed for the task with comprehensive medical knowledge.

However, the development of an EHR retriever has been severely limited by the lack of training

data (Jin et al., 2023; Zhao et al., 2023). The required query-document relevant pairs were traditionally accessible only through manual annotation. The prohibitive costs of such annotations inevitably constrain the dataset scale to only dozens of queries, and the resulting models perform barely on par with BM25 (Soni and Roberts, 2020). There have been attempts to generate large-scale relevance judgments automatically using string match algorithms or Large Language Models (LLMs) (Shi et al., 2022; Gupta et al., 2024). The increase in dataset scale leads to significant improvements in model performance. Yet, the queries used in these works are still provided by human experts or fixed vocabularies, limiting the scale and diversity of the training data. Consequently, the models lack generalizability and are only effective for specific diseases or even particular queries.

In this work, we aim to develop a series of **D**ense **R**etrieval models for **E**lectronic **H**ealth **R**ecord, dubbed **DR.EHR**. Specifically, to address the need for extensive medical knowledge and generalizable models, we propose a two-stage training pipeline based on MIMIC-IV discharge summaries (Johnson et al., 2023). In the first stage, we extract medical entity mentions from the EHRs and perform massive knowledge injection using a biomedical KG. In the second stage, inspired by Doc2Query (Nogueira et al., 2019), we utilize LLMs to generate relevant entities for each EHR to collect large-scale and diverse training data. The training data collection pipeline is summarized in Figure 1.

We train two variants of `DR.EHR`, with 110M and 7B parameters, respectively, using contrastive learning with in-batch negatives. On CliniQ, `DR.EHR-small` significantly outperforms all existing dense retrievers including 7B models, while our 7B variant demonstrates further improvement, achieving state-of-the-art results on the benchmark. Detailed analysis demonstrates that the superiority of `DR.EHR` is substantial and consistent across different match types and query types. Specifically, it achieves near-perfect performance on string matching and exhibits notable improvements on the most challenging semantic matching, such as implication and abbreviation matching. Through extensive ablation studies, we validate the effectiveness of each component in the training pipeline, further substantiating the model's enhanced medical knowledge and adaptability to EHR retrieval tasks.

Our contributions can be summarized as follows:

- We propose a two-stage training pipeline that leverages knowledge injection and synthetic data, addressing the lack of medical knowledge in models and diverse training data of large scale.
- We develop and release `DR.EHR`, a series of state-of-the-art dense retrieval models specifically designed for the task of EHR retrieval. To the best of our knowledge, `DR.EHR` is the first dense EHR retrieval model that is generalizable to a wide range of medical entities.
- A detailed analysis demonstrates that `DR.EHR` overcomes the limitations of general-domain dense retrievers, exhibiting significantly richer medical knowledge and enhanced semantic matching capabilities.

2 Related Work

2.1 EHR retrieval

Most EHR retrieval methods rely on exact matches and heavily leverage biomedical KGs (Hanauer et al., 2015; Ruppel et al., 2020). One popular approach to utilizing KGs for EHR retrieval is to identify medical entities in the EHRs and then match these entities with user queries (Bonacin et al., 2018; Goodwin and Harabagiu, 2017). Other systems use KGs for query expansion. By incorporating synonyms, abbreviations, and related concepts of user queries, these methods can significantly improve the recall rate (Zhu et al., 2013; Alonso and Contreras, 2016). However, these methods are limited to exact matching and fixed vocabularies, and therefore struggle to process complex EHRs.

Constrained by the shortage of training data, only a limited number of studies have explored the application of supervised learning and language models in this field. Soni and Roberts (2020) utilized the data from TREC Medical Record tracks to train a BERT-based re-ranker. However, only about 65 queries were used for training, and the resulting model barely performed on par with BM25. Shi et al. (2022) employed string matching to annotate the training data on imaging reports, and trained a dense retriever based on SentenceBERT (Reimers and Gurevych, 2019). Despite its superiority on the leave-out test set, only hundreds of queries were incorporated, all focused on searching for diseases and anatomical findings in imaging reports. Therefore, their model lacks generality. Recently, Gupta et al. (2024) trained the Onco-Retriever series using

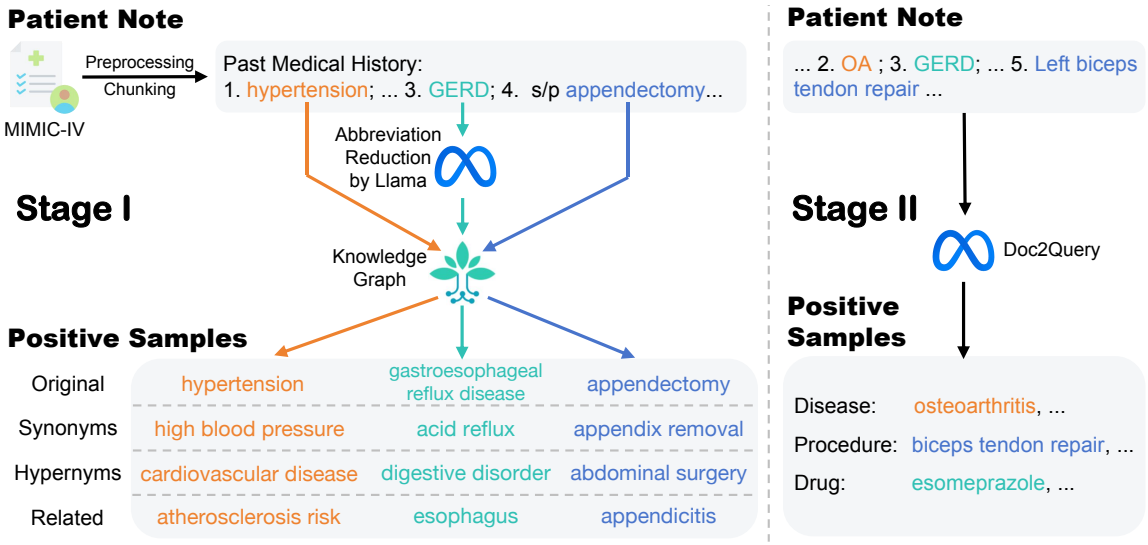

Figure 1: The training data collection pipeline of the two stages. In the first stage (left), the positive samples are defined as string-matched entities, reduced abbreviations, and their synonyms, hypernyms, and related entities sourced from the KG. In the second stage (right), the positive samples are generated by an LLM using Doc2Query. Note: OA is an abbreviation for osteoarthritis, and esomeprazole is generated since it is commonly used to treat GERD.

a private dataset and annotations based on GPT-3.5, with model parameter sizes of 500M and 2B. On the manually annotated test set, Onco-Retriever outperformed the properitary model developed by OpenAI and `SFR-Embedding-Mistral` (Rui Meng, 2024), which is based on Mistral 7B (Jiang et al., 2023). Yet, they only used 13 queries related to oncology, severely limiting the model's range of application. Clearly, there is a lack of an EHR retriever that can effectively address the semantic match challenge and be applied to a wide range of queries.

2.2 Knowledge injection

Knowledge injection has been widely adopted as an effective approach to enriching the models' knowledge in the biomedical domain, primarily through KGs (Trajanov et al., 2022). Knowledge injection can be performed either during the pre-training phase or during fine-tuning for downstream tasks. Hao et al. (2020) incorporated a relationship prediction task constructed from UMLS, the most widely used biomedical KG, to enhance the model's medical capabilities. Michalopoulos et al. (2020) also utilized UMLS and introduced UmlsBERT. By enhancing the model with semantic types of the entities and an additional prediction task for related entities, UmlsBERT demonstrated improvements across a variety of clinical tasks. Others focus on obtaining better entity representations via knowledge injection and language models (Yuan et al., 2020; Ying et al., 2024). CODER (Yuan et al., 2020) employed contrastive learning on terms and relation triplets from UMLS to improve term normalization, significantly outperforming existing medical embeddings. Similarly, Liu et al. (2020) introduced SapBERT, which used metric learning to cluster synonyms and achieved state-of-the-art results in medical entity linking tasks.

Knowledge injection has also been applied to dense retrieval. Tan et al. (2023) fed an additional entity embedding sequence into the BERT model and used an entity similarity loss to inject knowledge into the model. The resulting model, ELK, outperformed general domain retrievers in zero-shot biomedical retrieval tasks by a large margin.

2.3 Synthetic data for retrieval

Synthesizing data for retrieval may be traced back to Doc2Query (Nogueira et al., 2019), which was further expanded by Cheriton (2019). The idea behind these methods was to generate pseudo queries for documents as document expansion. With the rapid development of dense retrieval, training data soon became a scare resource, and research on synthetic data for retrieval turned to generate relevant queries from documents for model training. Dai et al. (2022) utilized the FLAN model (Wei et al., 2021) to generate pseudo queries for each of the BEIR (Thakur et al., 2021) datasets. Wang

et al. (2023) leveraged proprietary LLMs to generate diverse synthetic data across hundreds of thousands of tasks and 93 languages. In the biomedical domain, Xu et al. (2024) also relied on proprietary LLMs and generated synthetic data for biomedicine. So far, there has been no attempt to apply synthetic data for EHR retrieval.

3 Methods

We use MIMIC-IV discharge summaries as our training corpus. Following Zhao et al. (2025), we first clean the notes by removing all masks and excessive punctuation, and by converting all text to lowercase. Then, we split all patient records into 100-word chunks with overlap of 10 words. Based on this training corpus, we propose a two-stage training pipeline with synthetic data specifically designed for EHR retrieval. The overall training data collection pipeline along with an example is demonstrated in Figure 1.

3.1 Stage I: Knowledge injection pre-training

In the first stage, we aim to enrich the model's medical knowledge through contrastive learning. Specifically, for each note chunk used as an anchor, we first identify all entity mentions from it that are indexed in BIOS (Yu et al., 2022), the largest biomedical KG to date[1], as the initial positive sample set.

Then, to further enhance the model's abilities to identify abbreviations, we prompt `Llama-3.1-8B-Instruct`[2] to perform abbreviation reduction, and include the full names of the abbreviations appearing in the note as additional positive samples. We conduct several cleaning steps to remove any noise generated by the LLM, ensuring that the cleaned full names appear in BIOS. The prompt used for abbreviation reduction and the detailed cleaning process are described in Appendix A.

Finally, as the core step to inject knowledge from the KG, we look up each positive entity in BIOS and incorporate their synonyms, hypernyms (*is_a* relationship), and related entities (other relationships such as *may_treat* and *may_cause*) into the positive sample set. We do not include hyponyms (*reverse_is_a* relationship) since the information contained in the note is insufficient to deduce the hyponyms, and they will not be considered relevant in the downstream retrieval task.

In summary, given an anchor note chunk, its positive sample set consist of string-matched entities, full names of reduced abbreviations, and additional terms incorporated through BIOS.

3.2 Stage II: Synthetic data fine-tuning

In the second stage, we aim to fine-tune the model to optimize for the downstream EHR retrieval task using synthetic data. Following CliniQ, we also focus on the task of entity retrieval, and consider three types of query entities: diseases, clinical procedures, and drugs. We use `Llama-3.1-8B-Instruct` to generate various types of entities separately and combine them as the positive samples. For better semantic matching capabilities, we prompt the LLM to generate entities that are either explicitly mentioned in or can be implicitly inferred from each note chunk. The prompts used are provided in Appendix B.

3.3 Model training

We train two models of different sizes: `DR.EHR-small`, a BERT-based encoder with 110M parameters, initialized from `bge-base-en-v1.5`[3] (Xiao et al., 2023); and `DR.EHR-large`, a 7B decoder using the Mistral (Jiang et al., 2023) architecture, initialized from `NV-Embed-v2`[4]. These initialization choices are due to the superior performance of these models within their respective parameter sizes.

With different model architectures, the two models use distinct pooling strategies. For `DR.EHR-small`, we take the `[CLS]` embedding from the last layer as the text representation. For `DR.EHR-large`, we adopt last token pooling. The similarity $S(i, j)$ for an anchor i and a sample j is calculated as the cosine similarity of the two text embeddings.

In both stages, we train the model using Multi-Similarity Loss (MSL, Wang et al., 2019) with in-bach negatives. Formally, given an anchor i, its positive samples $\mathcal{P}(i)$, and its negative samples $\mathcal{N}(i)$, MSL first defines informative samples as follows:

$$\mathcal{P}'(i) = \{j | j \in \mathcal{P}(i), S(i,j) < \max_{k \in \mathcal{N}(i)} S(i,k) + \epsilon\} \quad (1)$$

[1] We also tried UMLS, which yielded suboptimal results.

[2] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[3] https://huggingface.co/BAAI/bge-base-en-v1.5

[4] https://huggingface.co/nvidia/NV-Embed-v2

$$\mathcal{N}'(i) = \{j | j \in \mathcal{N}(i), S(i,j) > \min_{k \in \mathcal{P}(i)} S(i,k) - \epsilon\} \tag{2}$$

where ϵ is a hyperparameter. The loss for each anchor is calculated as follows:

$$\mathcal{L} = \frac{\log(1 + \sum_{j \in \mathcal{P}'(i)} \exp(-\alpha(S(i,j) - \lambda)))}{\alpha} + \frac{\log(1 + \sum_{j \in \mathcal{N}'(i)} \exp(\beta(S(i,j) - \lambda)))}{\beta} \tag{3}$$

where α, β, and λ are hyperparameters. In our experiments, we use $\epsilon = 0.1, \alpha = 2, \beta = 50$, and $\lambda = 0.5$, determined by grid search.

4 Experiments

4.1 Statistics of the training data

From the 332k discharge summaries in MIMIC-IV, we obtain over 5.8M note chunks, each 100 words long, for training, with an average of 17.5 chunks per note. In the first training stage, the positive samples for each note chunk comprise three parts, with entities added from the KG further divided into three types: synonyms, hypernyms, and related entities. For training efficiency, we only include at most two synonyms, two hypernyms, and two related entities for each positive entity sourced from string matching or abbreviation reduction. For each hypernym or related entity included, we also incorporate at most one synonym. Consequently, for each positive entity, we add up to 10 terms from the KG. In our pilot study, adding more entities did not lead to significant improvement. Detailed statistics of these positive samples are presented in Table 1. On average, each note chunk is associated with 137.9 positive samples, resulting in a total of over 802M samples. Hypernyms, with an average of 50.9 samples per chunk, contribute the most, followed by related entities (38.6) and synonyms (30.2). Abbreviations account for the smallest proportion, with only 2.4 reduced abbreviations per chunk. Notably, nearly 28% of chunks have no positive samples from this source.

In the second training stage, the number of positive samples generated is significantly less than in the first stage. Detailed statistics, categorized by entity type, are provided in Table 2. On average, each chunk has 15.8 positive samples generated by the LLM, resulting in a total of nearly 86M samples. The generated entities exhibit a relatively even distribution among the three entity types.

Table 1: Statistics of positive samples for each chunk used in the first training stage. Avg: average; Q1: first quartile; Q3: third quartile; KG: knowledge graph.

Source	Avg	Q1	Q3	Max	Sum
String Match	15.7	12	20	64	91M
Abbreviation	2.4	0	3	25	14M
KG					
Synonym	30.2	22	38	127	176M
Hypernym	50.9	38	64	185	296M
Related	38.6	25	51	216	225M
Overall	137.9	102	172	588	802M

Table 2: Statistics of positive samples for each chunk used in in the second training stage. Avg: average; Q1: first quartile; Q3: third quartile.

Entity Type	Avg	Q1	Q3	Max	Sum
Disease	5.4	3	7	33	26M
Procedure	7.3	5	9	31	42M
Drug	4.6	2	6	32	20M
Overall	15.8*	11	20	63	86M

* The LLM may generate repeated entities in three rounds so the combined count is less than the sum of three types.

4.2 Model training

In our experiments, the maximum token length is set to 512 for note chunks and 16 for entities. To facilitate batch training, we up-sample or down-sample the positive entities of each chunk to a fixed number. We employ distinct data allocation strategies for the two models across two training stages, due to the different GPU memory requirements of the models and the varying dataset scales for each stage. The detailed hyperparameters are presented in Table 3. `DR.EHR-large` is trained with less data due to the higher GPU memory constraints.

The models are trained using 8 Nvidia A800 GPUs. Following Lee et al. (2024), `DR.EHR-large` is trained using low-rank adaptation (LoRA, Hu et al., 2021) with rank 16, alpha 32 and a dropout rate of 0.1. To further reduce GPU memory requirements, techniques including Bfloat 16 training and DeepSpeed ZeRO-2 are applied to `DR.EHR-large`. All training processes are optimized using AdamW (Loshchilov and Hutter, 2017) with default parameters and a learning rate of 1e-4. We set a warmup ratio of 0.1 and a linear decay for the learning rate scheduler.

Table 3: Data-related hyperparameters used for different models across different training stages. Pos: the number of positive samples per chunk.

Stage	Model	Pos	Batch Size*	Epoch
I	small	128	32	3
	large	32	16	1
II	small	16	32	1
	large	16	16	1

* With in-batch negatives, the ratio of positive to negative samples is batch size minus one.

4.3 Model evaluation

We evaluate our models on CliniQ, a comprehensive and publicly available EHR retrieval benchmark of large scale. CliniQ is constructed with 1k patient summaries from MIMIC-III, split into 16.5k chunks of 100 words each. It contains over 1k queries of three types: diseases, clinical procedures, and drugs, collected from structured codes in MIMIC and annotated by GPT-4o. It incorporates two retrieval settings: Single-Patient retrieval where models are tasked with ranking the chunks of a single patient note given a query, and Multi-Patient retrieval, where model are required to retrieve relevant chunks from the entire set of 16.5k chunks. On Single-Patient Retrieval, models are evaluated with Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (NDCG), and Mean Average Precision (MAP). On Multi-Patient Retrieval, models are evaluated with MRR, NDCG at 10, and recall at 100. CliniQ provides additional semantic match assessment by further classifying the relevance judgments into various categories.

4.4 Main results

The performance of DR.EHR on CliniQ is presented in Table 4, in comparison with bge-base-en-v1.5, MedCPT[5] (Jin et al., 2023), text-embedding-3-large by OpenAI, gte-Qwen2-7B-Instruct[6] (Li et al., 2023), and NV-Embed-v2. Our proposed models demonstrate superior performance on CliniQ. Specifically, DR.EHR-small with 110M parameters outperforms all existing dense retrievers, including the proprietary embedding model by OpenAI and state-of-the-art 7B models, by a remarkable margin. The large variant with 7B parameters demonstrate further significant improvement on Multi-Patient Retrieval. The advantages of DR.EHR are consistent and substantial across both retrieval settings and all metrics. Notably, we improve the MAP on Single-Patient Retrieval from the previous SOTA of 80.21 to 89.12 for DR.EHR-small and 88.92 for DR.EHR-large, and the Recall@100 on Multi-Patient Retrieval from the previous SOTA of 51.54 to 64.11 for DR.EHR-small and 67.20 for DR.EHR-large.

[5] https://huggingface.co/ncbi/MedCPT-Article-Encoder

[6] https://huggingface.co/Alibaba-NLP/gte-Qwen2-7B-instruct

5 Analysis

5.1 Semantic match assessment

The detailed results of the semantic match assessment in CliniQ are presented in Table 5. For brevity, we only report the average scores of MRR, NDCG, and MAP. DR.EHR demonstrates significant improvements over the baseline models. Specifically, DR.EHR addresses the challenge of insufficient exact match capabilities observed in general domain dense retrievers (Zhuang et al., 2023) in the context of EHR retrieval, achieving near-perfect performance on the string match benchmark in CliniQ. In terms of semantic matches, DR.EHR-small outperforms its initialization model by more than 10% across all categories, with a notable improvement of over 26% in abbreviation matching. These substantial gains underscore the effectiveness of the proposed pipeline. Through extensive knowledge injection and meticulously synthesized data, the models have learned to capture deep semantic associations between terms and represent them effectively in their embeddings.

5.2 Query type assessment

The detailed results for different query types (disease, procedure, and drug) are presented in Table 6. We additionally include the BM25 baseline, which achieves the best performance for drug searches in Multi-Patient Retrieval. The superiority of BM25 on this benchmark may be attributed to the fact that most drug queries consist of single words that appear verbatim in the notes. DR.EHR demonstrates consistent and significant improvements across all query types. Notably, it addresses the limitations of other dense retrievers in drug matching, improving the average scores by 12% and 24% in the two retrieval settings, respectively.

Table 4: Performance of various dense retrievers on CliniQ. QE: Query expansion. Dim: Dimension of the embeddings. R@100: Recall at 100.

Model	Size	Dim	Single-Patient			Multi-Patient		
			MRR	NDCG	MAP	MRR	NDCG@10	R@100
bge-base-en-v1.5	110M	768	82.48	83.59	74.54	54.97	56.51	39.50
MedCPT	220M*	768	84.23	85.49	77.42	47.21	50.07	41.97
text-embedding-3-large	-	3072	85.16	86.09	78.36	59.54	60.45	48.75
gte-Qwen2-7B-Instruct	7B	3584	84.59	85.33	77.02	60.39	62.06	48.04
NV-Embed-v2	7B	4096	86.57	87.36	80.21	59.48	62.06	51.54
DR.EHR-small	110M	768	92.96	93.26	89.12	67.06	68.75	64.11
w/o stage I	110M	768	91.61	92.00	87.15	65.55	67.59	60.42
DR.EHR-large	7B	4096	93.01	93.19	88.92	68.95	71.32	67.20

* MedCPT has separate query encoder and document encoder, so we count the parameter size as the summation of both models.

Table 5: Performance of various dense retrievers and ablation study on Single-Patient Retrieval, dissected by match types. The score for each type is the average of MRR, NDCG, and MAP. In the ablation study part, "w/o stage I" indicates the removal of stage I training, and each row starting with "+" represents adding extra training data in stage I to the previous row, with the same training data split as in Table 1.

Model	String	Synonym	Abbreviation	Hyponym	Implication
bge-base-en-v1.5	86.75	71.57	57.15	64.42	52.75
NV-Embed-v2	87.34	83.28	72.13	75.07	59.96
DR.EHR-small	97.34	86.01	83.37	76.88	67.56
w/o stage I	97.27	82.13	78.31	71.06	63.91
+ String Match	97.60	81.37	78.26	70.23	63.18
+ Abbreviation	97.47	81.69	80.40	69.98	63.96
+ KG–Synonym	97.66	84.07	80.79	71.31	64.35
+ KG–Hypernym	97.42	85.87	81.86	75.71	64.19
DR.EHR-large	97.59	86.26	85.08	74.96	65.32

Table 6: Performance of various retrieval methods and ablation study for different query types. The score for each type is the average of MRR, NDCG, and MAP in Single-Patient Retrieval, and the average of MRR, NDCG@10, and Recall@100 in Multi-Patient Retrieval. In the ablation study part, "Stage I +" indicates using only the specific type of synthesized data for training during stage II.

Model	Single-Patient			Multi-Patient		
	Disease	Procedure	Drug	Disease	Procedure	Drug
BM25	64.69	64.81	72.08	33.76	33.55	76.91
bge-base-en-v1.5	75.98	75.83	82.47	40.46	41.48	62.06
NV-Embed-v2	81.95	82.82	86.04	51.50	54.11	63.76
DR.EHR-small	87.52	83.95	94.61	51.58	50.90	86.03
Stage I + Disease	72.04	64.77	77.09	49.60	44.49	60.15
Stage I + Procedure	70.98	68.29	89.75	45.15	48.86	80.75
Stage I + Drug	59.97	57.10	90.62	33.28	32.14	85.49
DR.EHR-large	75.04	70.79	94.13	54.37	52.65	88.89

5.3 Ablation study

We conduct three ablation studies using DR.EHR-small. First, we ablate the stage I training and present the results in Tables 4 and 5. The results demonstrate that the knowledge injection phase significantly contributes to the final performance of DR.EHR, particularly on Recall@100 for Multi-Patient Retrieval. Detailed analysis of different match types reveals that this contribution is primarily attributed to semantic matches. The knowledge injection phase improves model performance by approxiamately 5% across all semantic match types.

To gain a deeper understanding of the contributions of knowledge injection, we divide the Stage I training data into five parts, as shown in Table 1, and sequentially incorporate each part to demonstrate their individual effects. The results, presented in Table 5, demonstrate that each portion of the training data significantly enhances performance on the corresponding benchmark, confirming that DR.EHR effectively acquires extensive knowledge from KGs. Notably, the additional training data also improves performance on other types of matching in most cases, indicating enhanced generalizability of DR.EHR.

For the second stage training, we divide the synthetic data according to the generated query types, and use them separately to train a series of models. As expected, results in Table 6 demonstrates that synthetic data tailored to specific query types improves model performance on the corresponding benchmark. Surprisingly, however, combining various types of synthetic data further enhances model capabilities significantly across all query types compared to models trained on individual data types. This synergistic effect of "1+1+1>3" might suggest that our model benefits from transfer learning during the second stage of training. When exposed to diverse query types, DR.EHR learns to capture broader semantic patterns and deeper knowledge connections, resulting in enhanced generalization capabilities and improved learning efficiency.

5.4 Case study

We conduct several case studies comparing bge-base-en-v1.5 and DR.EHR-small. For each match type, one example is selected, and the queries, note chunks, corresponding ranks, and cosine similarities generated by the two models are provided in Appendix C. The rank is calculated after excluding relevant chunks of other match types, and the cosine similarity is computed between the query and the relevant part (see Table 7) within the chunks. Our observations reveal that DR.EHR-small successfully identifies various types of matches, and its higher cosine similarities demonstrate its ability to learn extensive medical knowledge and represent information in clinical notes more effectively.

6 Conclusion

In this paper, we propose a two-stage training pipeline specifically designed for the task of EHR retrieval. The first stage employs KGs for knowledge injection through pre-training, while the second stage fine-tunes the model for the retrieval task with synthetic data generated by LLMs. Using this pipeline, we develop and release DR.EHR, a state-of-the-art EHR retriever available in two model sizes. Extensive experiments demonstrate that DR.EHR significantly outperforms baseline models across various settings, match types, and query types. Notably, our model exhibits exceptional capabilities in both string matching and semantic matching. Ablation studies confirm the contribution of each component in the training pipeline, underscoring its overall effectiveness.

7 Limitations

This study has several limitations. First, the evaluation of our model is restricted to a single benchmark, specifically the task of entity retrieval. The neglect of other query types, such as natural language questions and complex criteria, is due to the lack of publicly available benchmarks. We call for future efforts to construct richer and more diverse public benchmarks. Second, the quality of synthetic data in both stages of our work could be improved, as noise exists in both KGs and LLM-generated data. However, conducting data quality filtering on such a large scale is computationally intensive and exceeds our current resource constraints. Third, while hard negatives are known to significantly enhance model performance, particularly during task-specific fine-tuning (e.g., stage II training) (Karpukhin et al., 2020; Zeng et al., 2022), the design of synthetic hard negative data is non-trivial. We leave this challenge for future research.

References

Israel Alonso and David Contreras. 2016. Evaluation of semantic similarity metrics applied to the automatic retrieval of medical documents: An umls approach. Expert Systems with Applications, 44:386–399.

Rodrigo Bonacin, Júlio Cesar dos Reis, Edemar Mendes Perciani, and Olga Nabuco. 2018. Exploring intentions on electronic health records retrieval. studies with collaborative scenarios. Ingénierie des Systèmes d Inf., 23:111–135.

David R. Cheriton. 2019. From doc2query to doctttttquery.

Zhuyun Dai, Vincent Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. ArXiv, abs/2209.11755.

Tracy Edinger, Aaron M Cohen, Steven Bedrick, Kyle Ambert, and William Hersh. 2012. Barriers to retrieving patient information from electronic health record data: failure analysis from the trec medical records track. In AMIA annual symposium proceedings, volume 2012, page 180. American Medical Informatics Association.

Travis R. Goodwin and Sanda M. Harabagiu. 2017. Knowledge representations and inference techniques for medical question answering. ACM Transactions on Intelligent Systems and Technology (TIST), 9:1 – 26.

Shashi Kant Gupta, Aditya Basu, Bradley Taylor, Anai Kothari, and Hrituraj Singh. 2024. Onco-retriever: Generative classifier for retrieval of ehr records in oncology. Preprint, arXiv:2404.06680.

David A. Hanauer, Qiaozhu Mei, James Law, Ritu Khanna, and Kai Zheng. 2015. Supporting information retrieval from electronic health records: A report of university of michigan's nine-year experience in developing and using the electronic medical record search engine (emerse). Journal of biomedical informatics, 55:290–300.

Boran Hao, Henghui Zhu, and Ioannis C Paschalidis. 2020. Enhancing clinical bert embedding using a biomedical knowledge base. In 28th international conference on computational linguistics (coling 2020).

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. ArXiv, abs/2310.06825.

Qiao Jin, Won Kim, Qingyu Chen, Donald C Comeau, Lana Yeganova, W John Wilbur, and Zhiyong Lu. 2023. Medcpt: Contrastive pre-trained transformers with large-scale pubmed search logs for zero-shot biomedical information retrieval. Bioinformatics, 39(11):btad651.

Qiao Jin, Chuanqi Tan, Zhengyun Zhao, Zheng Yuan, and Songfang Huang. 2021. Alibaba damo academy at trec clinical trials 2021: Exploringembedding-based first-stage retrieval with trialmatcher. In Text Retrieval Conference.

Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. 2023. Mimic-iv, a freely accessible electronic health record dataset. Scientific data, 10(1):1.

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.

Bevan Koopman, Guido Zuccon, Peter Bruza, Laurianne Sitbon, and Michael Lawley. 2016. Information retrieval as semantic inference: A graph inference model applied to medical search. Information Retrieval Journal, 19:6–37.

Vojtech Lanz and Pavel Pecina. 2024. Paragraph retrieval for enhanced question answering in clinical documents. In Workshop on Biomedical Natural Language Processing.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. arXiv preprint arXiv:2405.17428.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. ArXiv, abs/2308.03281.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2020. Self-alignment pretraining for biomedical entity representations. In North American Chapter of the Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In International Conference on Learning Representations.

George Michalopoulos, Yuanxin Wang, Hussam Kaka, Helen Chen, and Alexander Wong. 2020. Umlsbert: Clinical domain knowledge augmentation of contextual embeddings using the unified medical language system metathesaurus. arXiv preprint arXiv:2010.10391.

Skatje Myers, Timothy A. Miller, Yanjun Gao, Matthew M. Churpek, Anoop M. Mayampurath, Dmitriy Dligach, and Majid Afshar. 2024. Lessons learned on information retrieval in electronic health records: A comparison of embedding models and pooling strategies. Journal of the American Medical Informatics Association : JAMIA.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pretraining. arXiv preprint arXiv:2201.10005.

Elsa Negro-Calduch, Natasha Azzopardi-Muscat, Ramesh S Krishnamurthy, and David Novillo-Ortiz. 2021. Technological progress in electronic health record system optimization: Systematic review of systematic literature reviews. International journal of medical informatics, 152:104507.

Rodrigo Nogueira, Wei Yang, Jimmy J. Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. ArXiv, abs/1904.08375.

Anusri Pampari, Preethi Raghavan, Jennifer J. Liang, and Jian Peng. 2018. emrqa: A large corpus for question answering on electronic medical records. In Conference on Empirical Methods in Natural Language Processing.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In Conference on Empirical Methods in Natural Language Processing.

Shafiq Rayhan Joty Caiming Xiong Yingbo Zhou Semih Yavuz Rui Meng, Ye Liu. 2024. Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog.

Halley Ruppel, Aashish Bhardwaj, Raj N Manickam, Julia Adler-Milstein, Marc Flagg, Manuel Ballesca, and Vincent X Liu. 2020. Assessment of electronic health record search patterns and practices by practitioners in a large integrated health care system. JAMA network open, 3(3):e200512–e200512.

Luyao Shi, Tanveer F. Syeda-Mahmood, and Tyler Baldwin. 2022. Improving neural models for radiology report retrieval with lexicon-based automated annotation. In North American Chapter of the Association for Computational Linguistics.

Sarvesh Soni and Kirk Roberts. 2020. Patient cohort retrieval using transformer language models. AMIA ... Annual Symposium proceedings. AMIA Symposium, 2020:1150–1159.

Jiajie Tan, Jinlong Hu, and Shoubin Dong. 2023. Incorporating entity-level knowledge in pretrained language model for biomedical dense retrieval. Computers in biology and medicine, 166:107535.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. arXiv preprint arXiv:2104.08663.

Dimitar Trajanov, Vangel Trajkovski, Makedonka Dimitrieva, Jovana Dobreva, Milos Jovanovik, Matej Klemen, Alevs vZagar, and Marko Robnik-vSikonja. 2022. Review of natural language processing in pharmacology. Pharmacological Reviews, 75:714 – 738.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. ArXiv, abs/2401.00368.

Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. 2019. Multi-similarity loss with general pair weighting for deep metric learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 5022–5030.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners. ArXiv, abs/2109.01652.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. Preprint, arXiv:2309.07597.

Ran Xu, Wenqi Shi, Yue Yu, Yuchen Zhuang, Yanqiao Zhu, May D Wang, Joyce C Ho, Chao Zhang, and Carl Yang. 2024. Bmretriever: Tuning large language models as better biomedical text retrievers. arXiv preprint arXiv:2404.18443.

Songchun Yang, Xiangwen Zheng, Yu Xiao, Xiangfei Yin, Jianfei Pang, Huajian Mao, Wei Wei, Wenqin Zhang, Yu Yang, Haifeng Xu, Mei Li, and Dongsheng Zhao. 2021. Improving chinese electronic medical record retrieval by field weight assignment, negation detection, and re-ranking. Journal of biomedical informatics, page 103836.

Cheng Ye, Bradley A Malin, and Daniel Fabbri. 2021. Leveraging medical context to recommend semantically similar terms for chart reviews. BMC Medical Informatics and Decision Making, 21(1):353.

Huaiyuan Ying, Hongyi Yuan, Jinsen Lu, Zitian Qu, Yang Zhao, Zhengyun Zhao, Isaac Kohane, Tianxi Cai, and Sheng Yu. 2025. Genie: Generative note information extraction model for structuring ehr data. Preprint, arXiv:2501.18435.

Huaiyuan Ying, Zhengyun Zhao, Yang Zhao, Sihang Zeng, and Sheng Yu. 2024. Cortex: contrastive learning for representing terms via explanations with applications on constructing biomedical knowledge graphs. Journal of the American Medical Informatics Association, page ocae115.

Sheng Yu, Zheng Yuan, Jun Xia, Shengxuan Luo, Huaiyuan Ying, Sihang Zeng, Jingyi Ren, Hongyi Yuan, Zhengyun Zhao, Yucong Lin, K. Lu, Jing Wang, Yutao Xie, and Heung yeung Shum. 2022. Bios: An algorithmically generated biomedical knowledge graph. ArXiv, abs/2203.09975.

Zheng Yuan, Zhengyun Zhao, and Sheng Yu. 2020. Coder: Knowledge-infused cross-lingual medical term embedding for term normalization. Journal of biomedical informatics, page 103983.

Sihang Zeng, Zheng Yuan, and Sheng Yu. 2022. Automatic biomedical term clustering by learning fine-grained term representations. In Workshop on Biomedical Natural Language Processing.

Yichi Zhang, Tianrun Cai, Sheng Yu, Kelly Cho, Chuan Hong, Jiehuan Sun, Jie Huang, Yuk-Lam Ho, Ashwin N Ananthakrishnan, Zongqi Xia, et al. 2019. High-throughput phenotyping with electronic medical record data using a common semi-supervised approach (phecap). Nature protocols, 14(12):3426–3444.

Zhengyun Zhao, Qiao Jin, Fangyuan Chen, Tuorui Peng, and Sheng Yu. 2023. A large-scale dataset of patient summaries for retrieval-based clinical decision support systems. Scientific Data, 10.

Zhengyun Zhao, Hongyi Yuan, Jingjing Liu, Haichao Chen, Huaiyuan Ying, Songchi Zhou, and Sheng Yu. 2025. Evaluating entity retrieval in electronic health records: a semantic gap perspective. Preprint, arXiv:2502.06252.

Dongqing Zhu, Stephen T Wu, James J. Masanz, Ben Carterette, and Hongfang Liu. 2013. Using discharge summaries to improve information retrieval in clinical domain. In Conference and Labs of the Evaluation Forum.

Shengyao Zhuang, Linjun Shou, Jian Pei, Ming Gong, Houxing Ren, G. Zuccon, and Daxin Jiang. 2023. Typos-aware bottlenecked pre-training for robust dense retrieval. Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region.

A Details of Abbreviation Reduction

The prompt used for abbreviation reduction is provided in Figure 2. After reducing abbreviations, we conduct the following cleaning steps to eliminate potential noise generated by the LLM:

1. We remove abbreviations that do not appear in the original note.
2. We remove full names that are identical to their abbreviations.
3. We remove full names that are not indexed in BIOS.
4. We remove abbreviations that are only one character long.

The Prompt for Abbreviation Reduction

```
Replace the abbreviations of medical entities with their full names in the
clinical note below. For one abbreviation, only output once unless it refers
to different full names in the note. If no abbreviation is found in the note,
please output "NA". Otherwise, output in the following format (only output
the terms and nothing else):
###[abbreviation]
***[full name]
...

For example:
###ct
***computed tomography

###wbc
***while blood cell

Now the task begins. Here is the note:
{note}
```

Figure 2: The prompt used for abbreviation reduction. {note} is the placeholder for the note to be processed.

The Prompt for Synthetic Data Generation

```
{note}

Briefly summarize the {entity_type} explicitly mentioned or that can be
implicitly inferred from the medical record above. Only output the entity
names (in their standardized terms) in a list. Do not output the reasons.

Output format:
- Entity 1
- Entity 2
...
```

Figure 3: The prompt used for synthetic data generation. {note} is the placeholder for the note to be processed, and {entity_type} takes on the values of diseases, clinical procedures, and drugs.

B Prompt for synthetic data generation

The prompt used for synthetic data generation is given in Figure 3.

C Case studies

We present several cases in Table 7 where `bge-base-en-v1.5` fails to retrieve the relevant chunk, while `DR.EHR` succeeds. One example is provided for each match type.

Table 7: Case studies of the performance of DR.EHR compared to bge-base-en-v1.5 on Singel-Patient Retrieval. The last two columns are the rank of the corresponding chunk and the cosine similarity given by the two models. The rank is calculated after removing relevant chunks of other match types. The cosine similarity is between the query and the relevant part (in red).

Match Type	**Query**	**Patient note**	**bge**	**DR.EHR**
String	ceftriaxone	... She was given Vanc, **Ceftriaxone**, Flagyl, 2L IVF, and started on levophed ...	12 / 1.00	1 / 1.00
Synonym	phenytoin	... MEDICINE Allergies: **Dilantin**[1]...	7 / 0.61	1 / 0.86
Abbreviation	hypertension	...Past Medical History: (1) **HTN**[2](2) ...	15 / 0.61	1 / 0.89
Hyponym	interruption of the vena cava	... Prophylaxis: **IVC filter**[3]and Pneumoboots. ...	5 / 0.59	1 / 0.61
Implication	diabetes mellitus	... Medications on Admission: lipitor 40mg po qday **metformin**[4]1000mg po bid ...	11 / 0.66	2 / 0.86

[1] Dilantin is a brand name of phenytoin.
[2] HTN is the common abbreivation for hypertension.
[3] IVC filter is a subtype of interruption of the vena cava.
[4] Metformin is a common hypolycemic agent.