Synthetic Training of Retrievers (SynTR)

Github Source: https://github.com/twgjr/SynTR

ABSTRACT

I present SynTR, a pipeline that leverages large vision–language models (VLMs) to generate synthetic training data, specifically pseudo-queries and relevance judgments, from an unlabeled visual document corpus, and then fine-tunes a dense visual retriever via LoRA for improved retrieval performance. Building on the LARMOR framework for text retrieval, I adapt pseudo-query generation, multi-retriever ranking, and VLM-based judging to the DocVQA BeIR subset of the VIDORE benchmark. I compare a variety of dataset generation settings and fine-tune the Metric-AI/colqwen2.5-3b multilingual retriever. Despite modest gains over the base model, our results demonstrate the feasibility of unsupervised fine-tuning for private visual corpora.

1 INTRODUCTION

Retrieval-augmented generation (RAG) systems for visual document collections often rely on pretrained retrievers that struggle under domain shift when no labeled queries exist. LARMOR demonstrated that large language models (LLMs) can generate pseudo-queries and query relevance labels (qrels) to select the best text retriever in an unsupervised manner [1]. However, visual documents present new challenges: whole-page images are larger, and incorporating image context into prompts is resource-intensive. I ask: Can I not only select but also fine-tune a visual retriever using only synthetic data generated by a VLM, on a private unlabeled corpus?

I simulate a scenario where an entity such as a private company, medical institution, or government has a private document image collection without any queries or judgments. I adapt LARMOR's query-generation and judging steps for vision–language, generate a minimal supervised dataset, and fine-tune via LoRA [3]. The contributions are:

1. SynTR pipeline: an end-to-end method to generate pseudo-queries/pseudo-qrels and fine-tune a dense visual retriever.
2. Efficient judgments: only the necessary hard positives and hard negatives are generated per query, reducing VLM calls.
3. Prompt comparison: evaluation of both LARMOR-style (specific) and "generic" prompts for pseudo-query generation.

4. Empirical study: performance on the DocVQA subset of VIDORE [2], with analysis of gains and limitations.

2 RELATED WORK

2.1 Dense Retriever Selection & LARMOR

LARMOR leverages LLMs to generate synthetic queries and pseudo-relevance to rank text retrievers in an unsupervised setting [1].

2.2 Visual Document Retrieval

Visual Document Retrieval benchmarks like ViDoRe and models such as ColPali embed page images directly, achieving strong end-to-end retrieval but require supervised data for fine-tuning [2].

2.3 LoRA

LoRA for Efficient Adaptation enables parameter-efficient fine-tuning of large models in low-GPU settings, widely used for both text and vision–language models [3].

3 BACKGROUND

3.1 VIDORE DocVQA Subset

I focus on the DocVQA BeIR dataset: 500 scanned pages with corresponding questions, repurposed for retrieval (each question as query, its page as the relevant document). It is among the weakest-performing VIDORE subsets [2], making it ideal to measure fine-tuning impact.

3.2 Base Retriever Selection

In Phase 2, using LARMOR adapted for vision–language, I generated pseudo queries and judged them to rank several pretrained visual retrievers, confirming that Metric-AI/colqwen2.5-3b-multilingual was the top-ranked model for this dataset [4].

4 PROBLEM FORMULATION

Given an unlabeled image corpus D and no queries, the goal is to:

1. Generate a synthetic training set $\{(q, d^+, \{d^-\})\}$.
2. Fine-tune the selected base retriever R_0 on this data via LoRA.
3. Evaluate the fine-tuned retriever R^*against R_O using nDCG@10 on a held-out split.

Key challenges include prompt design for pseudo-queries (q), efficient positive and negative sampling ($d^+ \; and \; d^-$), and resource constraints for VLM-based judging.

A key step in the pipeline is generating a full and high-quality set of not only pseudo queries but also the relevance judgments. As seen in Figure 1, queries are generated from a subset of all documents. Then a retriever ranks all the documents against each query. Based on that ranking a judge determines the relevance of a subset of the ranking. This reveals many more potential strong positives and strong negatives.

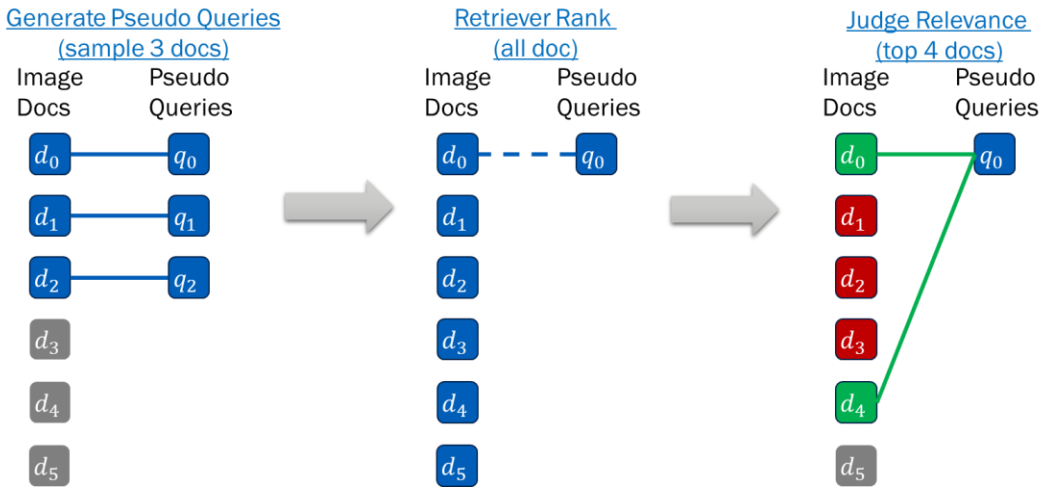


Figure 1. Conceptual Diagram of Dataset Generation

5 METHODOLOGY

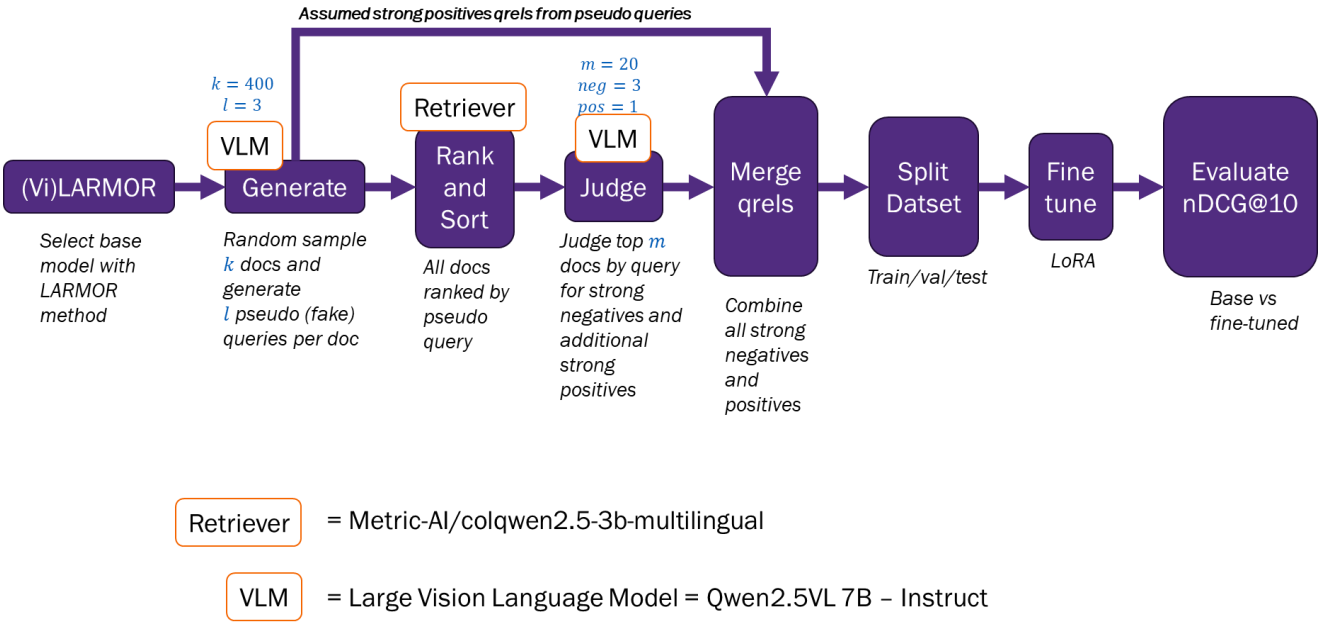


Figure 2. Overview of SynTR.

5.1 Pipeline Overview:

As shown in Figure 2, our pipeline proceeds through the following steps:

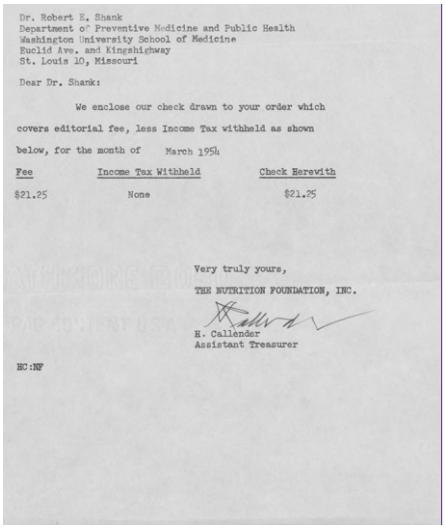
1. Base model selection: Use (Vi)LARMOR to pick the best pretrained retriever on the unlabeled corpus.
2. Pseudo-Query Generation: Randomly sample k documents and, for each, generate l pseudo-queries with a VLM, yielding strong positive qrels.
3. Ranking & Sorting: Embed and rank all documents per pseudo-query using the selected base retriever. The documents are ranked in descending order of similarity to the pseudo query.
4. VLM-Based Judging: For each query, judge the top m ranked docs to collect up to neg new hard negatives and pos hard positives.
5. Merge Qrels: Combine pseudo positives and judged positives and negatives into training tuples with at least one positive and varying number of negatives. A training sample has the structure of (query, positive, negatives).
6. Dataset Split: Shuffle and split into train/val/test.
7. Fine-Tuning: Adapt the base retriever with LoRA using contrastive CE loss and early stopping. I kept the LoRA parameters consistent for each modification of the training set generations to ensure fair comparability.

8. Evaluation: Compute nDCG or other ranking metric on the held-out test set for both base and fine-tuned models.

5.2 Pseudo-Data Generation

To generate the synthetic training data, I begin by sampling images from the corpus. For each image, I generate candidate queries from a VLM using one of two prompts. The first follows a "LARMOR-style" instruction. The prompts generates a somewhat unnatural pseudo query for a retrieval task. Queries are worded such that a person is already holding the document.

I introduce a new version which is more open-ended as shown in Figure 3 below.


Dr. Robert E. Shank
Department of Preventive Medicine and Public Health
Washington University School of Medicine
Euclid Ave. and Kingshighway
St. Louis 10, Missouri

Dear Dr. Shank:

We enclose our check drawn to your order which covers editorial fee, less Income Tax withheld as shown below, for the month of March 195[illegible]

Fee	Income Tax Withheld	Check Herewith
$21.25	None	$21.25

Very truly yours,

THE NUTRITION FOUNDATION, INC.

H. Callender
Assistant Treasurer

HC:NF

- LARMOR Style (Specific to the exact document)
- Prompt: "Generate a question that the following image can answer. Avoid generating general questions."
- Pseudo Query: What is the amount of the check enclosed in the letter?

- SynTR Style (More Generic)
- Prompt: "Imagine a user is researching a topic and is looking for documents that could provide helpful background or support. Given the image, generate a search query that would naturally lead a retrieval system to include this document. The query should reflect a topic of interest, not necessarily a direct fact the image contains. Present the query in the form of natural questions. Do not make any commentary about the search query. Only respond with the query."
- Pseudo Query: "documents about nutrition foundation payment processing history"

Figure 3. Pseudo Query Prompts

Following query generation, both the pseudo-queries and the full set of document images are embedded with the base retriever. Every document is ranked by its similarity to each query and the top m candidates are selected for relevance judgement by VLM.

In the judging phase, I again leverage the VLM to assess relevance: each query–document pair is classified as "Highly Relevant," "Somewhat Relevant," or "Not Relevant." I continue this process until a predefined number of positive and hard negatives are identified; the original page from pseudo-query generation also serves as a strong positive. Pairs labeled "Highly Relevant" become hard positives, while those labeled "Not Relevant" become hard negatives. This step is crucial for assembling a diverse set of query–document pairs beyond what the initial pseudo-query step provides.

One important caveat concerns hard-negative selection. Because only the top m ranked documents are examined in descending similarity order, some negatives may lie very close

in the embedding space to true positives. In practice, this may make it difficult for the model to separate positives from negatives in the contrastive training loss.

6 EXPERIMENT SETUP

All experiments were conducted using NVIDIA A100 GPUs with 80 GB of memory, leveraging mixed-precision (bfloat16) training and an 8-bit optimizer to enable memory-efficient fine-tuning of the 3-billion parameter Metric-AI/colqwen2.5-3b-multilingual retriever. I used typical settings for LoRA as well as those aligning with the base models original training. A parameter-efficient LoRA configuration was used with a rank of 128 and dropout rate of 0.1 applied to all attention and feedforward layers. Fine-tuning was performed with a learning rate of 2×10^{-4} and an effective batch size of 8, using gradient accumulation and early stopping to avoid overfitting. Each training run included up to 10 epochs, with the best model checkpoint selected based on nDCG@10 with the validation set.

To build the synthetic training dataset, I sampled 400 documents from a 500-page corpus and used a vision–language model to generate three pseudo-queries per document. These queries were then embedded along with the entire corpus using the selected base retriever, and the top 20 candidate documents for each query were then judged. A vision–language model judge was applied to these ranked results to identify additional strong positives and hard negatives based on semantic relevance. Each final training example consisted of one strong positive and three hard negatives, with variations introduced in how these were selected depending on the experimental condition.

To evaluate the impact of different design choices in the dataset generation pipeline, I defined five experimental configurations. These varied by the style of pseudo-query prompts (either a LARMOR-inspired, specific prompt or a more general open-ended prompt), the number of hard negatives (either one or three), the method of selecting positive examples (solely using the pseudo-query's source document or supplementing with VLM-judged positives), and how hard negatives were identified (either randomly selected from non-positive candidates or chosen by the VLM judge from top-ranked candidates). All other components of the pipeline, including the total number of queries, document sampling strategy, embedding method, LoRA configuration, and data split ratios (80% training, 10% validation, 10% test), were kept consistent to isolate the effect of each individual change.

The evaluation process involved computing nDCG@10 on the test split for both the base retriever and the best-performing fine-tuned checkpoint, ensuring a fair comparison across configurations. This allowed me to observe how improvements in dataset quality,

especially through enhanced relevance judgment and prompt design, translated into measurable gains in retrieval performance.

7 RESULTS

Table 1. Retrieval performance (nDCG@10) for various dataset generation settings.

Query Set	# Hard Negatives	Hard Positive Method	Hard Negative Method	nDCG@10		% change
				Base	Fine-Tuned	
Specific	3	Pseudo Query Only	Random Non-Positive	**0.952**	0.932	-2.1%
General	1	Pseudo Query Only	VLM Judged	**0.882**	0.875	-0.8%
General	3	Pseudo Query Only	Random Non-Positive	0.869	**0.873**	0.5%
General	3	Pseudo Query Only	VLM Judged	0.882	**0.899**	1.9%
General	3	Pseudo Query + VLM Judged	VLM Judged	0.871	**0.893**	2.5%

Table 1 shows a summary of the results for each version of the dataset used to fine-tune the base model. It demonstrates a progressive improvement as the dataset parameters were changed. In general, it appears that the following dataset generation settings are optimal for improving fine-tuning performance:

1. Using VLM generated relevance judgements
2. Using more hard negatives generated by the VLM
3. Using a more general purpose pseudo query generation prompt
4. Including additional hard positive from VLM relevance judgments.

The progressive improvements are likely due to increasing the number of high-quality training examples. I believe it is mainly due to the VLM relevance judgements. It could be viewed as a bipartite graph with queries on the one side and documents on the right. The VLM is essentially filling in more of the edges between queries and documents that are more likely to be correct. This is opposed to the simpler yet faster approach of only using initial pseudo query generations as the only strong positives and randomly choosing from the remaining documents assuming they are negatives. Additionally, adding more relevance judgements added more training examples overall.

8 LIMITATIONS

Several factors limit the performance of our approach.

First, the retriever I fine-tuned, ColQwen2.5 with 3 billion parameters, may lack the model capacity to significantly reshape its embedding space under a lightweight LoRA adapter. Larger models might adapt better.

Second, as discussed in Section 5, selecting negatives strictly based on descending similarity can cause issues. Hard negatives can lie close to positives in embedding space, making it harder for the model to learn to separate them during contrastive training.

Third, the relevance judging step was constrained by resource limitations. I judged only a fixed number of top-ranked documents per query, meaning some highly relevant documents further down the list were likely missed.

Fourth, the DocVQA dataset is notably challenging even for state-of-the-art models. In the ViDoRe benchmark, the best models only achieved around 67.4% nDCG@10. Our pseudo-generated datasets scored higher, suggesting that our synthetic data may not fully capture the subtle complexities of the original task.

Finally, prompt engineering remains a crucial dependency. Different unlabeled corpora may require significant prompt adjustments to generate useful pseudo-queries aligned with document content.

9 FUTURE WORK

There are several directions for improving the SynTR pipeline. One opportunity is to scale up model capacity by fine-tuning larger retrievers, such as models with 7 billion parameters or more. Larger models may be better able to adapt their embedding spaces under lightweight fine-tuning techniques like LoRA, potentially leading to greater performance gains. Another improvement involves expanding the relevance judgment window. Rather than judging only the top-ranked candidates for each query, sampling positives and negatives from a broader range could reduce the risk of selecting overly difficult negatives that are almost indistinguishable from true positives.

In addition to expanding the judgment window, incorporating more positive examples could strengthen the training dataset.

Finally, there is an opportunity to model the retrieval task as a bipartite graph between queries and documents, where edges represent relevance relationships. Applying graph-based machine learning techniques could help identify additional relevant documents that

may not be discovered through synthetic judgments alone. Together, these improvements target both the quality of the synthetic data and the model's capacity to learn from it, offering promising paths to further close the gap between unsupervised and supervised retrieval performance.

10 CONCLUSION

SynTR demonstrates that unsupervised fine-tuning of a visual retriever using synthetic data is not only feasible but also leads to consistent, if modest, improvements. By carefully designing the pseudo-query and relevance judgment process, even lightweight fine-tuning methods like LoRA can move the needle on retrieval performance.

There's still a lot of room to grow. Scaling up to larger models, expanding the relevance judgment window, and incorporating more strong positives are all promising directions to further boost performance. Exploring graph-based approaches for modeling the retrieval space could also open up new possibilities. SynTR is just the beginning, and the early results are encouraging for unsupervised retrieval on private visual corpora.

REFERENCES

1. E. Khramtsova, S. Zhuang, M. Baktashmotlagh, and G. Zuccon. 2024. Leveraging LLMs for Unsupervised Dense Retriever Ranking. SIGIR ’24.
2. M. Faysse, H. Sibille, T. Wu, B. Omrani, G. Viaud, C. Hudelot, and P. Colombo. 2025. ColPali: Efficient Document Retrieval with Vision Language Models. ICLR ’25.
3. E. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, L. Wang, W. Chen, and Y. L. S. Wang. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv preprint arXiv:2106.09685v2.
4. T. Green. 2025. Phase 2 Report – “Implementation of ‘Leveraging LLMs for Unsupervised Dense Retriever Ranking’ for Vision–Language.”