DocReRank: Single-Page Hard Negative Query Generation for Training Multi-Modal RAG Rerankers

Navve Wasserman[1], Oliver Heinimann[1], Yuval Golbari [1], Tal Zimbalist [1]
Eli Schwartz[2], Michal Irani [1]
[1]Weizmann Institute of Science [2] IBM Research Israel
navve.wasserman@weizmann.ac.il

Abstract

Rerankers play a critical role in multimodal Retrieval-Augmented Generation (RAG) by refining ranking of an initial set of retrieved documents. Rerankers are typically trained using hard negative mining, whose goal is to select pages for each query which rank high, but are actually irrelevant. However, this selection process is typically passive and restricted to what the retriever can find in the available corpus, leading to several inherent limitations. These include: limited diversity, negative examples which are often not hard enough, low controllability, and frequent false negatives which harm training. Our paper proposes an alternative approach: *Single-Page Hard Negative Query Generation*, which goes the other way around. Instead of retrieving negative pages per query, we generate hard negative queries per page. Using an automated LLM-VLM pipeline, and given a page and its positive query, we create hard negatives by rephrasing the query to be as similar as possible in form and context, yet *not* answerable from the page. This paradigm enables fine-grained control over the generated queries, resulting in diverse, hard, and targeted negatives. It also supports efficient false negative verification. Our experiments show that rerankers trained with data generated using our approach outperform existing models and significantly improve retrieval performance[1].

1 Introduction

Accurately retrieving relevant documents is fundamental to many natural language processing (NLP) tasks. Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) is a widely adopted framework in which models retrieve external evidence to guide generation. This enables scaling to large document collections while maintaining factual grounding. In real-world applications, *multimodal RAG* extends this framework beyond plain text to include visual and structural elements such as figures, tables, and full-page document images.

While first-stage retrieval models (Karpukhin et al., 2020; Khattab and Zaharia, 2020; Xiong et al., 2020) aim to identify a small set of relevant candidates, their reliance on embedding similarity often limits precision, especially in visually complex settings. To improve fine-grained relevance, a second-stage *reranker* is commonly used to reorder the top-k documents based on richer query-document interaction. Reranking has been extensively studied in text-based RAG (Nogueira et al., 2019, 2020; Sun et al., 2023; Liu et al., 2025), with one prominent approach adapted to the multimodal setting (Chaffin and Lac, 2024).

A common training strategy is hard negative mining: for each query, passages or pages labeled as negatives are selected based on their relevance ranking from a retrieval model. However, this approach faces several key limitations: (i) ***Limited hard negatives:*** Negatives are restricted to documents in the corpus, limiting diversity and difficulty; (ii) ***Uncontrollable:*** The selection process is passive; only what the retriever pulls out can be used, making it hard to target specific model weaknesses (e.g., fine-grained distinctions); (iii) ***Computationally expensive:*** The process requires embedding the entire corpus and performing a full retrieval search for each query, making it resource-intensive; (iv) ***False negatives:*** Documents incorrectly labeled as irrelevant despite containing the answer are common and can significantly harm training.

We propose an inverse approach: ***Single-Page Hard Negative Query Generation***. Instead of retrieving negative pages per query, we generate hard queries per a given document page. This approach is inverse not only because it generates rather than retrieves, but also because the negatives are queries instead of documents, avoiding the need to synthesize full pages, which is far more complex and often low quality. Our automated pipeline combines Large Language Models (LLMs) and Vision

[1]For Datasets & Model see: https://navvewas.github.io/DocReRank/.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 8640–8658
November 4-9, 2025 ©2025 Association for Computational Linguistics

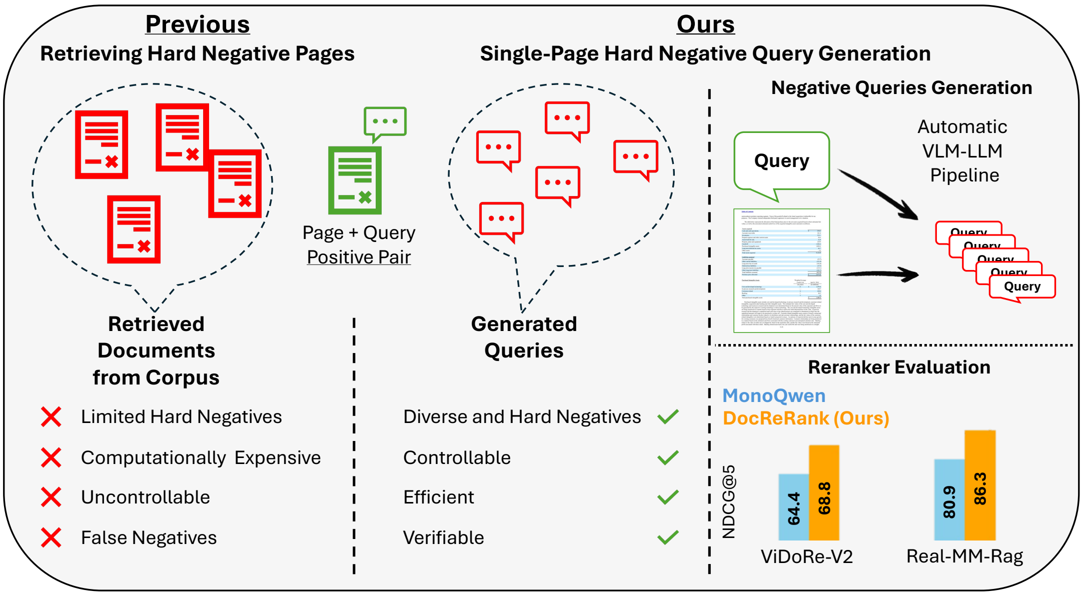


Figure 1: **Proposed Single-Page Hard Negative Query Generation Approach.** While previous approaches retrieve hard negative pages per query from a document corpus, our method goes the other way around: We generate hard negative queries per page using an automated LLM-VLM pipeline. Our reranker, “DocReRank” which trains on this kind of data, outperforms models trained with document-based hard negatives.

Language Models (VLMs) to generate positive queries which are answerable from the page, then rephrases them into hard negatives that are structurally and semantically similar but unanswerable.

This approach addresses the key limitations of document-focused hard negative mining: (i) ***Diverse and Hard Negatives:*** By generating queries instead of relying on retrieving documents, we avoid dataset constraints, and can produce diverse, challenging negatives for any page. (ii) ***Controllable:*** We explicitly control the type of negative queries generated, allowing us to target specific model weaknesses; (iii) ***Efficient:*** Our method eliminates the need to embed and search over large document corpora for each query, significantly reducing the computational cost of hard negative generation; (iv) ***Verifiable:*** Since multiple negative queries relate to the same page, VLM-based verification is fast and reliable, reducing false negatives.

We show that training rerankers with data generated by our proposed *Single-Page Hard Negative Query Generation* approach significantly outperforms models trained with document-based hard negatives alone. Furthermore, our method can be tailored to address specific model weaknesses. For example, we observed that rerankers perform poorly on financial documents and having recurring errors involving fine-grained factual distinctions (e.g., years, numerical values, entity names). Therefore, we curate a finance-focused dataset using targeted prompts that modify individual attributes during negative query generation. This produces especially challenging negatives that improve reranker robustness in structured, information-dense settings. While finance motivated this effort, such fine-grained variations also appear in other domains, like corporate reports and scientific papers, highlighting the broader applicability of our approach. Training with this dataset yields additional performance gains.

Lastly, we examine the impact of training data quality beyond initial query generation. In the original ColPali train-set, many positive queries closely mirror document wording, encouraging shallow keyword matching rather than true semantic understanding. Following the insights of Wasserman et al. (2025), we create a rephrased version of the dataset, modifying query phrasing while preserving meaning. Models trained on this data show improved performance on standard benchmarks and greater robustness on the rephrased version of the Real-MM-RAG benchmark.

Our contributions are as follows:

- We propose *Single-Page Hard Negative Query Generation* approach, for creating challenging, controllable, and verifiable hard negatives.
- *DocReRank*, a multimodal reranker that outperforms previous models across benchmarks.
- *ColHNQue*, a dataset (ColPali Hard Negative Queries) suitable for training rerankers.
- *FinHNQue*, a finance-focused negative queries dataset targeting fine-grained distinctions.

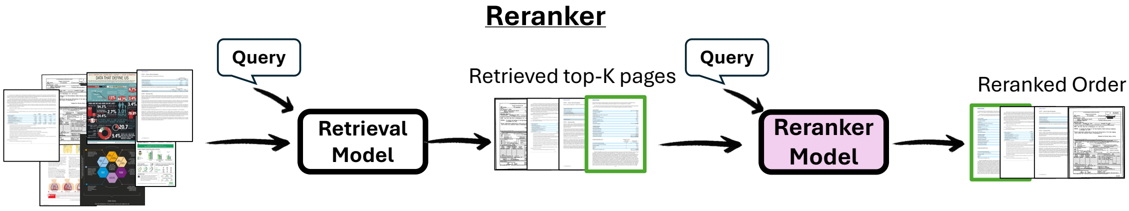


Figure 2: **Re-ranking Framework.** Given a query and a document corpus, a retrieval model first retrieves the top-K relevant pages. A reranker then reorders these K pages based on the query to improve retrieval quality.

2 Related Work

In modern information-retrieval systems, a first-stage retriever scans a large corpus to select a handful of candidate documents for a user's query, and a second-stage reranker then applies more costly models to reorder those far fewer candidates and boost precision.

2.1 Retrieval Models

Early retrieval methods such as TF–IDF (Sparck Jones, 1972) and BM25 (Robertson et al., 1994) relied on simple lexical matching. These approaches offer extreme efficiency but little semantic understanding. Transformer-based dense retrievers — BERT (Devlin et al., 2019), T5 (Raffel et al., 2020), and DPR (Karpukhin et al., 2020) — map queries and documents into continuous embeddings, dramatically boosting recall at the cost of higher compute. Hybrid retrievers like ColBERT (Khattab and Zaharia, 2020) and ANCE (Xiong et al., 2020) fuse token-level interactions with vector representations. Yet, text-only retrievers still struggle on richly formatted or visually complex documents. To bridge that gap, multimodal pipelines are needed. First approaches used captioning-based methods to translate visual elements into natural language (Ramos et al., 2023) or contrastive embeddings to align visual and textual features (Radford et al., 2021; Zhai et al., 2023). A more recent line of work leverages the strong capabilities of VLMs to analyze full document images by embedding entire pages, bypassing OCR-based extraction. Methods like VISRAG (Yu et al., 2024) and DSE (Ma et al., 2024) generate embeddings directly from document images. Similarly, ColPali (Faysse et al., 2024) produces multi-vector embeddings for ColBERT-style late interaction retrieval, using PaliGemma (Beyer et al., 2024), or in its ColQwen (Faysse et al., 2024) variant, Qwen2-VL (Wang et al., 2024). These approaches show clear improvements over earlier methods.

2.2 Reranking Models

A Reranker's tasks is to get the top-K candidate documents retrieved in the first stage, and output those documents in a new order, ranked by predicted relevance to the query. Rerankers can be grouped into three main types: *Pointwise* methods score each document independently given the query (e.g., MonoBERT (Nogueira et al., 2019), MonoT5 (Nogueira et al., 2020) and CEDR (MacAvaney et al., 2019)). *Pairwise* methods compare pairs of documents and predict which one is more relevant, as in DuoT5 (Pradeep et al., 2021). *Listwise* methods optimize over the entire ranked list to capture global ordering subtleties (RankGPT (Sun et al., 2023), PE-Rank (Liu et al., 2025)).

As with retrieval, *multimodal* reranking is needed to handle documents enriched with images, tables, and complex layouts. Specifically, as the best retrieval models operate directly on page images, these kinds of rerankers are necessary. While vision–language models can be adapted to judge query–page correspondence, this is far from an optimal solution. This field is in its early stages with with one prominent model, the MonoQwen (Chaffin and Lac, 2024) reranker, which employs LoRA to fine-tune the Qwen2.5-VL-7B-Instruct VLM using ColPali training data with hard negative mining.

Hard Negative Mining Hard negative mining is an important part of effective reranker training, involving the selection of challenging negative examples. A trained retrieval model fetches the top-K passages or pages per query, and those not labeled as positive are treated as negatives. By identifying difficult negative examples that are semantically similar to the query yet irrelevant, the model learn more discriminative features and improves the training quality. Early reranker training such as DPR (Karpukhin et al., 2020), BERT passage re-ranking (Nogueira and Cho, 2019), MonoBERT (Nogueira et al., 2019) and ColBERT (Khattab and Zaharia, 2020) relied on simple hard negatives, derived from static BM25-mined samples or in-

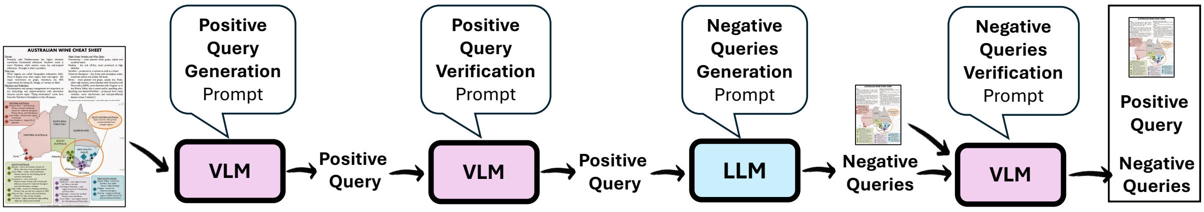

Positive Query Generation Prompt
VLM
Positive Query
Positive Query Verification Prompt
VLM
Positive Query
Negative Queries Generation Prompt
LLM
Negative Queries
Negative Queries Verification Prompt
VLM
Positive Query
Negative Queries

Figure 3: **Dataset Construction Pipeline.**

batch examples. Following works adopted dynamic and multi-retriever mining strategies (e.g., R^{2}anker Zhou et al., 2022) as well as positive-aware hard negative mining (Moreira et al., 2024) to further boost performance.

However, passage or document-level hard negative mining has several inherent limitations, including limited diversity, false negatives, stale negatives, and little control over the types of negatives retrieved. To address these limitations, we propose Single-Page Hard Negative Query Generation: a fully automated LLM and VLM pipeline that generates challenging queries for each document page, rather than retrieving hard negative documents for a given question. This paradigm enables fine-grained control, diverse and targeted negatives, and efficient false negative filtering via VLMs, since set of negative queries are associated with a single page and can be verified together. Overall, this produces more challenging and higher-quality training data.

3 Dataset Generation

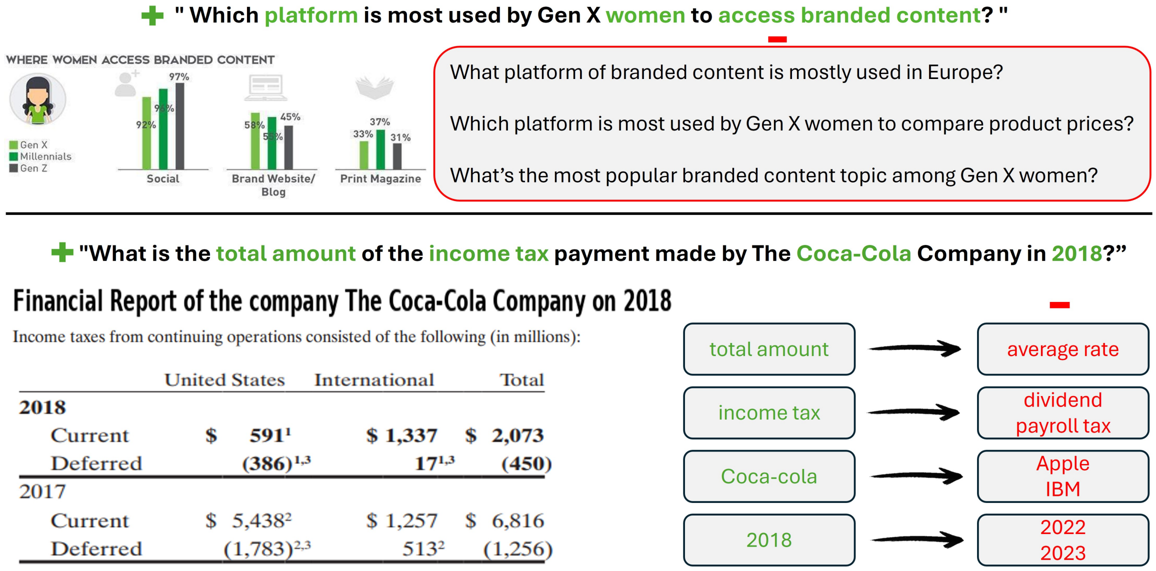
We propose a new approach for generating document page–query pairs using a dedicated Single-Page Hard Negative Query Generation strategy (see Fig. 3). Our full pipeline consists of four stages: the first two handle ***positive query generation and verification***, while the latter two focus on ***hard negative query generation and verification***. If only the hard negative generation is needed (e.g., to extend existing datasets), the process can begin from step 3. Below, we describe the full pipeline and later demonstrate how it can be adapted to model-specific weaknesses.

3.1 Generation Pipeline

Given an image of a document page, the goal is to produce both positive and hard negative queries that relate to the content of that page.

Positive Query Generation We adapt the prompt design from Wasserman et al. (2025) (see Fig. S1) and use the Pixtral-12B VLM (Agrawal et al., 2024) to generate N candidate positive queries per page. The prompt is designed to encourage RAG-style questions; natural questions that a user might ask without having seen the page itself. It further emphasizes multimodal understanding by focusing on page elements such as figures, tables, and diagrams. The second stage verifies that each generated query is answerable from the page content. We use the Qwen2.5-VL-7B-Instruct VLM with a dedicated prompt (see Fig. S2) to validate each query. This model is different from the one used for generation, reducing model-specific biases. After verification, we retain one validated query per page to form a clean set of (page image, positive query) pairs. While multiple positives could be used, we select a single one for simplicity and fair comparison to previous datasets.

Hard Negative Query Generation Given a page image and its corresponding positive query, our goal is to generate hard negative queries i.e., queries that are not answerable from the page, but are similar in structure and context to the positive, making them difficult for rerankers to distinguish. This process is divided into two distinct stages, generation and verification, as we found it significantly more effective to decouple the linguistic task of rephrasing a query from the visual task of grounding it in the page content. Specifically, it is relatively easy for an LLM to generate query variants that are semantically close to the original but seek different information, whereas asking a VLM to handle both rephrasing and verification often led to degraded quality in the resulting negatives. First, we use the Qwen2.5-7B-Instruct LLM to generate 12 variants of the positive query (see prompt in Fig. S3). These are designed to be similar in topic and form but seek different information. This LLM-only step is well-suited for understanding the instruction and generating plausible alternatives. Next, each candidate query is validated using the Qwen2.5-VL-7B-Instruct VLM. We input the document page along with each negative candidate using two slightly different verification prompts (see Fig. S2) to improve robustness. Only queries



+ " Which platform is most used by Gen X women to access branded content? "

−

What platform of branded content is mostly used in Europe?

Which platform is most used by Gen X women to compare product prices?

What's the most popular branded content topic among Gen X women?

+ "What is the total amount of the income tax payment made by The Coca-Cola Company in 2018?"

Financial Report of the company The Coca-Cola Company on 2018

Income taxes from continuing operations consisted of the following (in millions):

	United States	International	Total
2018			
Current	**$ 591[1]**	**$ 1,337**	**$ 2,073**
Deferred	**(386)[1,3]**	**17[1,3]**	**(450)**
2017			
Current	$ 5,438[2]	$ 1,257	$ 6,816
Deferred	(1,783)[2,3]	513[2]	(1,256)

−

total amount → average rate

income tax → dividend
payroll tax

Coca-cola → Apple
IBM

2018 → 2022
2023

Figure 4: **Examples of Our Generated Negative Queries.** We show examples of a cropped page and its positive query, along with the generated negative queries. Top: hard negatives generated using the general pipeline. Bottom: negatives generated using finance fine-detail prompts, which modify specific properties in the query.

that both prompts classify as unanswerable are kept as valid hard negatives. This automated pipeline yields high-quality triplets (page image, positive query, hard negatives), which can then be used to train reranker models more effectively.

3.2 Finance Focused Generation

A key advantage of our proposed negative query generation approach is its adaptability to specific document types. Although the commonly used ColPali training set contains a variety of financial documents, the model performance on financial benchmarks remains notably lower. We further observed from error analysis conducted on the Real-MM-RAG benchmark, that models have difficulties in handling fine-grained information distinctions (see Figs. S5 and S6).

While these issues are most prominent in financial documents, such fine-grained errors (e.g. confusing numerical values, time periods, or entity names) can also occur in other document types that include structured or data-rich content (e.g., restaurant annual reports or corporate filings). Therefore, improving robustness to small variations in factual details can benefit a wide range of use cases involving factual or financial information.

To address this, we developed a dedicated set of prompts (see Fig. S4) that, given a positive query, instructs the model to generate a variant by modifying exactly one property; such as the year (e.g., 2022 → 2024), company name (e.g., Apple → IBM), numerical value (e.g., price, percentage), financial metric (e.g., revenue, sales, acquisitions), subject metric (e.g., dividends, stocks, options), or business segment (e.g., cloud, software, manufacturing). This produces highly targeted hard negatives that challenge the model's ability to distinguish fine-grained but critical details.

We applied this method to the FinTabNet dataset (Zheng et al., 2021), which contains annual reports from S&P 500 companies, generating a training set of 20K pages paired with corresponding positive and domain-specific hard negative queries.

3.3 Rephrased Dataset

To improve model semantic understanding and robustness, we introduce a rephrased version of the ColPali training set. We rephrase 50% of the positive queries while preserving their meaning using an LLM. This encourages the model to rely on semantic understanding rather than surface-level cues.

4 DocReRank Training

Our *DocReRank* reranker is based on the pretrained Vision-Language Model Qwen2-VL-2B-Instruct (Wang et al., 2024), using the same Low-Rank Adaptation (LoRA) (Hu et al., 2022) configuration as in the ColPali paper (Faysse et al., 2024). LoRA is applied to the transformer layers of the LLM, while the visual encoder is kept frozen. The reranker is trained on triplets of (query, document page image, label), where the label is 1 if the image contains the answer to the query (positive),

Benchmark	Axa	Economics	Restaurant-rse	Restaurant-esg	Biomedical	Economics-ML	Restaurant-ML	Biomedical-ML	Avg
ColPali	55.0	53.4	51.5	55.2	57.8	47.6	52.5	55.6	53.6
Qwen-VLM	61.8 +6.8	47.9 -5.5	57.7 +6.2	60.8 +5.6	56.4 -1.4	49.8 +2.2	58.0 +5.5	56.1 +0.5	56.1 +2.5
MonoQwen	68.0 +13.0	57.8 +4.4	58.2 +6.7	68.5 +13.3	65.9 +8.1	56.7 +9.1	59.2 +6.7	62.6 7.0	62.1 +8.5
***DocReRank**-Base*	**71.8** +16.8	**64.7** +11.3	**58.3** +6.8	**68.8** +13.6	**68.0** +10.2	**58.3** +10.7	**61.2** +8.7	**62.9** +7.3	**64.2** +10.7
***DocReRank**-Full*	**70.7** +15.7	**67.5** +14.1	**63.4** +11.9	**71.8** +16.6	**66.8** +9.0	**62.8** +15.2	**63.3** +10.8	**63.7** +8.1	**66.2** +12.7
ColQwen	64.4	61.3	54.4	61.2	62.2	52.6	56.0	56.8	58.6
Qwen-VLM	66.1 +1.7	51.9 -9.4	**63.6** +9.2	64.2 +3.0	57.7 -4.5	50.4 -2.2	62.9 +6.9	56.3 -0.5	59.1 +0.5
MonoQwen	71.4 +7.0	60.3 -1.0	61.6 +7.2	**72.3** +11.1	66.7 +4.5	58.2 +5.6	61.8 +5.8	63.1 +6.3	64.4 +5.8
***DocReRank**-Base*	**77.3** +12.9	**67.5** +6.2	63.3 +8.9	71.7 +10.5	**68.5** +6.3	**60.7** +8.1	**65.2** +9.2	**63.8** +7.0	**67.2** +8.6
***DocReRank**-Full*	**75.3** +10.9	**68.7** +7.4	**66.6** +12.2	**75.8** +14.6	**68.2** +6.0	**64.7** +12.1	**66.0** +10.0	**64.7** +7.9	**68.8** +10.1

Table 1: **Model Performance on the ViDoReV2 Benchmark.** Retrieval NDCG@5 Results. The first row in each block shows first-step retrieval results using ColPali or ColQwen. The remaining rows correspond to second-step reranking results. Our model *DocReRank-Base* is trained with a similar configuration to MonoQwen but includes our generated data. *DocReRank-Full* is trained with generated fine-grained details and rephrased negative queries.

and 0 otherwise (negative). We follow the MonoQwen (Chaffin and Lac, 2024) training framework, where the model is prompted to generate the token "True" or "False" given a query and an image. During training, a softmax over the logits of these tokens provides a relevance score, used both as the loss and as the basis for reranking during inference.

4.1 Training Datasets

We use three types of training data: (i) standard hard-negative-mined data of document pages, (ii) data generated by our proposed Single-Page Hard Negative Query Generation approach , and (iii) Rephrasing Variants.

Document Page Hard Negative Mining (*Col-HNDoc*): We use a version of the ColPali training dataset, also used in MonoQwen, with hard negatives provided by Nomic-AI (Nomic AI, 2025) (MonoQwen hard negatives mining is not available). Each query is paired with one positive page and three hard negative pages sampled from the top-10 retrieval results. This yields approximately 120k positive pairs and 360k negative pairs, totaling around 480k training examples.

Single-Page Hard Negative Query Generation: Our generated datasets include two variants: (i) *Col-HNQue* (ColPali Hard Negative Queries) Based on the same ColPali training set, we keep the original query–positive page pairs and generate three hard negative queries for each page. This dataset is matched in size to *Col-HNDoc*, with the only difference being in the method of generating negatives. (ii) *Fin-HNQue* (Finance Hard Negative Queries): We apply our full query generation pipeline to 20k pages from the FinTabNet dataset (Zheng et al., 2021), generating one positive query and three hard negatives per page. This results in 80k training examples tailored to the financial domain and to fine-grained information distinctions (see section 3.2).

Rephrasing Variants: We further introduce an augmented versions of both *Col-HNDoc* and *Col-HNQue* (*Reph-*) where 50% of the positive queries are rephrased while preserving their meaning.

4.2 Training Procedure

All models were trained using the Hugging Face Trainer with a learning rate of 1e-4, 100 warm-up steps, and a learning rate decay schedule. Training was conducted on 4 NVIDIA L40S GPUs, with each GPU processing a batch size of 32 examples per step. Each batch consists of 8 positive (image, query) pairs and 24 corresponding negatives.

This batch structure ensures that each positive example is accompanied by its respective negatives within the same batch. For the *ColHNDoc* dataset, a positive consists of a query and its corresponding document page, while the negatives are three other document pages that are hard negatives for the same query. For our proposed datasets, each positive consists of a document page and a positive query, and the negatives are three hard negative queries generated for that page.

We use cross-entropy loss over the softmax probabilities of the "True" and "False" token logits. To address class imbalance between positives and negatives in each batch, we assign a weight ratio of 3:1 in favor of the positive examples. All models are trained for one epoch, with the number of training steps determined by the size of each dataset.

Benchmark	FinReport	FinSlides	TechReport	TechSlides	Avg
ColPali	52.9	62.7	80.4	89.4	71.4
Qwen-VLM	62.6 +9.7	77.0 +14.3	73.9 -6.5	78.8 -10.6	73.1 +1.7
MonoQwen	**73.0** +20.1	82.1 +19.4	79.4 -1.0	91.9 +2.5	81.6 +10.2
***DocReRank**-B*	71.6 +18.7	**86.3** +23.6	**89.6** +9.2	**94.1** +4.7	**85.4** +14.0
***DocReRank**-F*	**73.2** +20.3	**86.8** +24.1	**89.5** +9.1	**94.4** +5.0	**86.0** +14.6
ColQwen	60.8	58.7	84.4	91.2	73.8
Qwen-VLM	65.5 +4.7	72.1 +13.4	73.4 -11.0	79.1 -12.1	72.5 -1.3
MonoQwen	75.6 +14.8	76.2 +17.5	79.4 -5.0	92.3 +1.1	80.9 +7.1
***DocReRank**-B*	**76.8**+16.0	**80.7** +22.0	**90.0** +5.6	**94.7** +3.5	**85.6** +11.8
***DocReRank**-F*	**79.0** +18.2	**80.9** +22.2	**90.4** +6.0	**94.8** +3.6	**86.3** +12.5

Table 2: **Model Performance on the Real-MM-RAG Benchmark.** Retrieval NDCG@5 results of Rerankers after first step retrieval with ColPali and ColQwen.

5 Results

In this section, we demonstrate the effectiveness of our data generation framework for reranker training. We first describe the experimental setup in section 5.1, then show in section 5.2 that training with our generated data significantly outperforms strong baselines under comparable settings. We further show in section 5.2 that combining domain-specific data targeting reranker weaknesses and rephrased queries leads to additional improvements. Finally, in section 6 we present a comprehensive set of ablations, including the contribution of each dataset and analyses of other method components.

5.1 Experimental Setup

Benchmarks. We evaluate on two multimodal retrieval benchmarks that closely reflect real-world RAG use cases, featuring challenging and information-seeking queries. ***ViDoReV2*** (Illuin Technology, 2025): This benchmark includes 8 evaluation datasets, three of which are multilingual. Some queries have answers that span multiple pages, contributing to overall lower performance. ***Real-MM-RAG*** (Wasserman et al., 2025): This benchmark includes four high-difficulty evaluation set: FinReport, FinSlides, TechReport, and TechSlides. We also evaluate on the rephrased version of this benchmark, provided by the authors, to assess model robustness and true semantic understanding.

Evaluation Metric We report the standard NDCG@5 as the primary evaluation metric, measuring the quality of the top-ranked retrieved pages. Additional metrics, such as Recall@5 and NDCG@10, are reported in the Appendix (see section A.3).

Retrieval Models. To evaluate the reranker's impact, we use two strong retrieval models following the ColPali paper (Faysse et al., 2024) approach: ***ColPali-v1.2*** and ***ColQwen2-v1.0***. We first retrieved top-20 pages per each query in the evaluation dataset using those models and then used the rerankers for reordering those top-20 pages.

Baseline Rerankers. As multimodal reranking is still a developing field, we compare against the following strong and relevant baselines: ***Qwen-VLM*** uses the Qwen2-VL-2B-Instruct model with our standard reranking prompt but without any fine-tuning. ***MonoQwen*** is a fine-tuned reranker trained using the MonoQwen approach on the ColPali dataset. It uses the same base model (Qwen2-VL-2B-Instruct) and training objective as our reranker but is trained solely on hard-negative-mined document-level data, without query generation or adaptation to model weaknesses.

5.2 Main Results

In Tables 1 and 2, we report NDCG@5 reranking results after retrieving with both ColPali and ColQwen. Retrieval-only performance is also shown for reference. We first evaluate our base model, ***DocReRank-Base***, which fine-tunes Qwen2-VL using a combined training set: half with negative pages from traditional document-level hard negative mining (*Col-HNDoc*), and half with hard negative queries from our generation approach (*Col-HNQue*). This dataset includes 120K positive examples and 360K negatives. This setup allows a direct comparison to MonoQwen, which uses a similar architecture and training data but relies only on document-based negatives.

As shown in the results, *DocReRank-Base* achieves significant improvements over retrieval-only baselines (e.g., +8.6 points with ColQwen on ViDoReV2), and clearly outperforms MonoQwen,

Benchmark	FinReport Rephrased	FinSlides Rephrased	TechReport Rephrased	TechSlides Rephrased	Avg
ColQwen	41.8	31.1	67.2	78.0	54.5
Qwen-VLM	49.3 +7.5	49.0 +17.9	60.6 –6.6	73.7 –4.3	58.2 +3.7
MonoQwen	49.0 +7.2	50.7 +19.6	**73.0** +5.8	82.6 +4.6	63.8 +9.3
***DocReRank**-F* *(w/o Reph)*	**55.0** +13.2	**53.1** +22.0	72.5 +5.3	**83.1** +5.1	**65.9** +11.4
***DocReRank**-F*	**57.1** +15.3	**52.1** +21.0	**79.0** +11.8	**88.0** +10.0	**69.0** +14.5

Table 3: **Performance on the Rephrased Real-MM-RAG Benchmark.** Retrieval NDCG@5 results of Rerankers after first step retrieval with ColQwen.

Benchmark	Axa	Economics	Restaurant-rse	Restaurant-esg	Biomedical	Economics-ML	Restaurant-ML	Biomedical-ML	Avg
ColQwen	64.4	61.3	54.4	61.2	62.2	52.6	56.0	56.8	58.6
Qwen-VLM	66.1 +1.7	51.9 –9.4	63.6 +9.2	64.2 +3.0	57.7 –4.5	50.4 –2.2	62.9 +6.9	56.3 –0.5	59.1 +0.5
FT on Col-HNDoc	73.2 +8.8	64.4 +3.1	62.3 +7.9	66.1 +4.9	67.8 +5.6	56.5 +3.9	62.3 +6.3	62.8 +6.0	64.4 +5.8
***DocReRank**-Base*	**77.3** +12.9	67.5 +6.2	63.3 +8.9	71.7 +10.5	**68.5** +6.3	60.7 +8.1	65.2 +9.2	63.8 +7.0	67.2 +8.6
***DocReRank**-B w Fin*	**76.8** +12.4	66.8 +5.5	**67.5** +13.1	73.8 +12.6	60.2 –2.0	**68.9** +16.3	**66.5** +10.5	64.3 +7.5	68.1 +9.5
***DocReRank**-B w Fin&Reph*	73.6 +9.2	**67.9** +6.6	**68.1** +13.7	**74.8** +13.6	**70.3** +8.1	62.9 +10.3	**67.0** +11.0	**65.5** +8.7	**68.8** +10.2
***DocReRank**-Full*	75.3 +10.9	**68.7** +7.4	66.6 +12.2	**75.8** +14.6	68.2 +6.0	**64.7** +12.1	66.0 +10.0	**64.7** +7.9	**68.8** +10.2

Table 4: **Ablation Results on ViDoReV2 Benchmark.** Retrieval NDCG@5 Results. We compare a model fine-tuned only on document-based hard negatives (*FT on Col-HNDoc*) to our *DocReRank-Base*, which outperforms this baseline. Adding finance-, fine-detail-specific generated queries (*Fin-HNQue*), and rephrased data leads to further performance gains.

which as far as we know, differs only on the training data (document hard negatives only). On ColQwen, we observe gains of +2.8 on ViDoReV2 and +4.7 on Real-MM-RAG.

We also evaluate our full model, *DocReRank-Full*, which incorporates the Finance-Focused Generation dataset (*Fin-HNQue*) and rephrased positive queries. This leads to additional gains across both retrieval models and benchmarks, demonstrating the impact of adapting to model-specific weaknesses and requiring model sematic understanding.

To further highlight the role of rephrasing, we evaluate on the rephrased version of Real-MM-RAG. In Table 3, we compare our full model trained with and without rephrased positives. Results show that training with rephrased queries improves robustness, although some performance drop remains when evaluated on rephrased benchmarks, emphasizing the challenge of moving beyond shallow keyword matching toward true semantic understanding.

6 Ablations & Analysis

6.1 Generated Data Contribution

We aim to demonstrate two key points: (i) our data generation approach offers clear benefits over using only document-based hard negatives, and (ii) the individual contribution of each of our generated datasets. To isolate the impact of our data, we fine-tuned the same model used in *DocReRank*, under identical training settings, but only with the *Col-HNDoc* dataset (based on document hard negative retreival). As expected, this model achieved results similar to MonoQwen, which was trained in a comparable manner. As shown in Table 4, our *DocReRank-Base* model outperforms the model trained solely on *Col-HNDoc*, demonstrating the added value of our query generation approach. Adding the *Fin-HNQue* (Finance Hard Negative Queries) dataset leads to further improvements, and incorporating the rephrased dataset boosts performance even more.

Importantly, all *DocReRank-Base* models were trained using the same number of total training examples, sampled from different datasets (see section A.2 for details). The full model was trained with twice the number of examples. While it shows comparable results in this table, it achieved additional improvements with ColPali retrieval (see Table S7).

6.2 Generalization to Real-World Queries

We aim to demonstrate that our generated negative queries are diverse and support generalization to real-world query patterns. First of all, this is reflected in the reranker's strong performance on ViDoReV2, which contains real-world queries across multiple benchmarks. To further validate this, we sampled 10,000 positive queries from the ColPlaIi training set and 10,000 generated negatives, embedded them with BGE-M3 (Multi-Granularity, 2024), and visualized them using t-SNE (see Fig. S7). The generated negatives show a distribution closely aligned with the positives, indicating both diversity and realism. The Fréchet Distance between the two distributions is also very low (FD = 0.090), confirming statistical similarity. Since our method generates negatives by rephrasing positives, the diversity of positives is key. The ColPlaIi training set already provides a wide variety of realistic queries, and our approach can naturally adapt to any future dataset with similarly diverse inputs.

6.3 VLM Verification Evaluation

To assess the reliability of our VLM-based verification, we conducted a human evaluation to assess the rate of false negatives and false positives remaining after the VLM-based verification. Six annotators were presented with a document page and two to five negative queries (242 queries in total). For each query, they judged whether the page contained an answer. An equivalent evaluation was performed on our generated positive queries. The results indicate a low false negative rate of 8.2% and no false positives, validating the effectiveness of the VLM-based verification. Importantly, the small proportion of residual errors is unlikely to affect training performance, as the reranker can still learn robustly from the majority of correctly labeled examples.

7 Conclusions

Our work challenges the conventional reliance on document-level hard negative mining for reranker training by introducing a query-generation alternative. A core insight is that generation offers greater controllability and diversity than retrieval from a fixed document corpus. Query generation is also more practical, as queries are short and easy to control. Grounding query generation in a single document page, gives even more controllability — enabling generation of multiple, diverse negatives tailored to specific content and allowing efficient verification. This enables us to generate harder negatives, verify unanswerability, and target known model weaknesses. We further show how this controllability can be used to generate specific negatives that match model-specific weaknesses. It can also be adapted to application-specific needs. For example, ensuring a model distinguishes machine type when answering questions about manufacturing manuals, to avoid returning answers for the wrong machine.

Our results show that query-level generation is a strong alternative to document mining, yielding superior reranking performance when used alongside traditional negatives. We believe this framework can be extended and refined to provide valuable training data for future research and deployment.

8 Limitations

While our approach represents a significant step toward better hard negative examples and has been shown to improve reranker training, several limitations remain. *Query variability:* Positive and corresponding hard negative queries are generated using a pipeline of VLMs and LLMs. Despite careful prompt instructions, not the full query space used by a human might be exploited. *Query verification:* To verify the answerability of a given query, VLMs are used. Nevertheless, despite strategies such as double verification using two separate prompts, false negatives and positives can occur, potentially limiting quality of hard negatives. *Reranker Dependency:* The reranker step fully depends on the initially provided ranked subset of the full document corpus by the retrieval algorithm. If the true positive document of a query is not listed in the provided subset, the reranker will never be able to provide the true answer neither.

Acknowledgment: This project was funded by the European Union (ERC, MindReading, 101142115).

References

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, et al. 2024. Pixtral 12b. *arXiv preprint arXiv:2410.07073*.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. 2024. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*.

Antoine Chaffin and Aurélien Lac. 2024. Monoqwen: Visual document reranking.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2024. Colpali: Efficient document retrieval with vision language models. *arXiv preprint arXiv:2407.01449*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Illuin Technology. 2025. Vidore benchmark v2: Visual document retrieval benchmark.

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*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Qi Liu, Bo Wang, Nan Wang, and Jiaxin Mao. 2025. Leveraging passage embeddings for efficient listwise reranking with large language models. In *Proceedings of the ACM on Web Conference 2025*, pages 4274–4283.

Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhu Chen, and Jimmy Lin. 2024. Unifying multimodal retrieval via document screenshot embedding. *arXiv preprint arXiv:2406.11251*.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. Cedr: Contextualized embeddings for document ranking. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 1101–1104.

Gabriel de Souza P Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. 2024. Nv-retriever: Improving text embedding models with effective hard-negative mining. *arXiv preprint arXiv:2407.15831*.

Multi-Linguality Multi-Functionality Multi-Granularity. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

Nomic AI. 2025. Colpali queries mined 2025-03-21 by source.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv preprint arXiv:2101.05667*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Rita Ramos, Desmond Elliott, and Bruno Martins. 2023. Retrieval-augmented image captioning. *arXiv preprint arXiv:2302.08268*.

S Robertson, Steve Walker, Susan Jones, and MHB GATFORD. 1994. Okapi at 3. In *Proceedings of the 3rd Text REtrieval Conference (-3)*, pages 109–126.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Navve Wasserman, Roi Pony, Oshri Naparstek, Adi Raz Goldfarb, Eli Schwartz, Udi Barzelay, and Leonid Karlinsky. 2025. Real-mm-rag: A real-world multi-modal retrieval benchmark. *arXiv preprint arXiv:2502.12342*.

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*.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, et al. 2024. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. *arXiv preprint arXiv:2410.10594*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986.

Xinyi Zheng, Doug Burdick, Lucian Popa, Peter Zhong, and Nancy Xin Ru Wang. 2021. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context.

Winter Conference for Applications in Computer Vision (WACV).

Yucheng Zhou, Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Guodong Long, Binxing Jiao, and Daxin Jiang. 2022. Towards robust ranker for text retrieval. *arXiv preprint arXiv:2206.08063*.

A Appendix

A.1 Evaluation Details

All evaluations, both for retrieval models and rerankers, were conducted using the ColPali evaluation framework and the `mteb` package. For reranker evaluation, we first retrieved the top-20 pages per query using a given retrieval model. Then, for each query, the reranker computed a relevance score between the query and each of the top-20 retrieved pages. These pages were subsequently re-ordered according to the reranker's relevance scores, and evaluation metrics were computed on the newly ranked list.

A.2 Training Details

Our models were trained using data from the following datasets: (i) ***Col-HNDoc*** – document-level hard negative mining based on the ColPali training set; (ii) ***Col-HNQue*** – our hard negative query generation applied to the ColPali training set; (iii) ***Fin-HNQue*** – our finance-specific hard negative query dataset; (iv) ***Reph-Col-HNDoc*** and (v) ***Reph-Col-HNQue*** – rephrased variants of the above datasets. Below we detail the data used for training each model. For simplicity, we report the number of positive examples used (each paired with 3 negatives, totaling 4× the size in training samples).

- ***FT on Col-HNDoc*** – trained on the full 120k positives and 360k negatives from *Col-HNDoc* using document-based hard negatives.
- ***DocReRank-Base*** – trained on 60k positives from *Col-HNDoc* and 60k from *Col-HNQue*.
- ***DocReRank-B w/ Fin*** – trained on 60k from *Col-HNDoc*, 40k from *Col-HNQue*, and the full 20k from *Fin-HNQue*.
- ***DocReRank-B w/ Fin & Reph*** – trained on 60k from *Reph-Col-HNDoc*, 40k from *Reph-Col-HNQue*, and 20k from *Fin-HNQue*.
- ***DocReRank-Full*** – trained on 120k from *Reph-Col-HNDoc*, 120k from *Reph-Col-HNQue*, and 20k from *Fin-HNQue*.

A.3 Additional Results

In the main paper, we focused on reporting the NDCG@5 metric. In Tables S1 to S6, we provide results for additional metrics, including Recall@1, Recall@5, and NDCG@10. We also report additional ablation results in Table S7, using ColPali retrieval, complementing the ColQwen-based ablations shown in Table 4.

A.4 Generation and Verification Prompts

Our multi-step query generation pipeline combines a Large Language Model (LLM) and a Vision-Language Model (VLM), each guided by specific prompts tailored to different stages of the process. We provide here the full prompts used in each step. The positive query generation prompt, shown in Fig. S1, is used with the VLM (Pixtral-12B) to generate natural, information-seeking queries that are answerable from the given document page. The two verification prompts, shown in Fig. S2, are used with the VLM (Qwen2.5-VL-7B-Instruct) to determine whether a query is answerable from the page content. Using two slightly different prompt formulations improves verification robustness. The generic hard negative query generation prompt, shown in Fig. S3, instructs the LLM to rephrase a given positive query into unanswerable variants that are similar in form and topic. Finally, the finance-specific prompt, shown in Fig. S4, focuses on fine-grained factual attributes (e.g., years, amounts, company names) to create particularly challenging negative queries for detail-sensitive content.

A.5 Licensing and General Information

All models and datasets used in this work comply with their respective licenses. Qwen2-VL (ColQwen2) and Qwen are licensed under Apache 2.0, with adapters released under the MIT license. PaliGemma (ColPali) follows the Gemma license, also with adapters under MIT. Pixtral-12B-2409 (mistralai) and Mixtral-8x22B are both released under the Apache 2.0 license, which permits unrestricted use, modification, and distribution. LLaMA 3.3 70B is released under the LLaMA 3.3 Community License Agreement.

All datasets used are in English, except for ViDoRe V2, which includes queries and documents in French. The ColPali training set includes subsampled academic datasets redistributed under their original licenses. It also incorporates synthetic datasets generated from publicly available internet content and VLM-generated queries, which are released without usage restrictions. The REAL-MM-RAG benchmark is distributed under the Community Data License Agreement – Permissive, Version 2.0 (CDLA-Permissive-2.0). The FinTabNet dataset is composed of data collected from publicly available sources. An AI assistant (ChatGPT) was used for minor grammar and sentence structure edits.

Benchmark	**Axa**	**Economics**	**Restaurant-rse**	**Restaurant-esg**	**Biomedical**	**Economics-ML**	**Restaurant-ML**	**Biomedical-ML**	**Avg**
ColPali	18.4	8.7	21.5	38.3	32.6	9.0	22.1	32.0	22.8
Qwen-VLM	**36.3**	6.0	27.1	35.6	30.5	7.1	26.3	29.8	24.8
MonoQwen	31.8	9.4	27.4	**49.7**	37.0	8.5	26.5	36.5	28.4
DocReRank-*Base*	**37.2**	**13.9**	**30.4**	**50.5**	**40.9**	**11.5**	**32.6**	**37.6**	**31.8**
DocReRank-*Full*	**37.2**	**12.8**	**33.6**	48.6	**39.2**	**13.0**	**33.3**	**37.2**	**31.9**
ColQwen	29.1	5.9	22.3	41.9	36.7	6.4	24.6	33.1	25.0
Qwen-VLM	**37.2**	7.6	27.7	34.3	30.1	7.5	27.8	29.9	25.3
MonoQwen	**31.8**	9.5	28.1	48.4	36.8	9.1	26.7	36.9	28.4
DocReRank-*Base*	**37.2**	**13.1**	**33.5**	**49.1**	**40.9**	**11.4**	**34.6**	**38.2**	**32.2**
DocReRank-*Full*	**37.2**	**12.3**	**33.3**	**50.2**	**38.7**	**13.2**	**33.0**	**37.5**	**31.9**

Table S1: **Performance on the ViDoReV2 Benchmark recall@1**

Benchmark	**Axa**	**Economics**	**Restaurant-rse**	**Restaurant-esg**	**Biomedical**	**Economics-ML**	**Restaurant-ML**	**Biomedical-ML**	**Avg**
ColPali	58.5	27.1	54.6	60.0	61.3	24.3	56.5	58.7	50.1
Qwen-VLM	58.0	27.2	**59.6**	66.9	59.9	26.9	**61.6**	59.7	52.1
MonoQwen	**64.6**	30.4	59.0	69.1	**68.6**	**31.5**	**61.7**	**64.3**	56.0
DocReRank-*Base*	64.3	**34.0**	58.2	**70.3**	**67.9**	30.5	60.7	63.4	**56.2**
DocReRank-*Full*	**65.1**	**36.1**	**60.4**	**72.3**	67.4	**32.6**	**61.6**	**65.0**	**57.6**
ColQwen	59.0	**35.2**	56.6	66.1	64.0	29.3	58.4	59.4	53.5
Qwen-VLM	58.0	27.2	**67.2**	71.1	62.2	26.9	**66.2**	60.4	54.9
MonoQwen	66.5	31.6	64.6	**73.6**	**70.2**	31.5	65.7	**64.9**	**58.6**
DocReRank-*Base*	**68.7**	**35.8**	61.8	72.5	68.6	**31.8**	63.9	64.6	58.5
DocReRank-*Full*	**67.8**	**35.8**	**66.1**	**77.6**	**69.4**	**34.1**	**66.1**	**66.3**	**60.4**

Table S2: **Performance on the ViDoReV2 Benchmark recall@5**

Benchmark	**Axa**	**Economics**	**Restaurant-rse**	**Restaurant-esg**	**Biomedical**	**Economics-ML**	**Restaurant-ML**	**Biomedical-ML**	**Avg**
ColPali	54.7	52.2	54.7	58.9	61.5	47.5	55.9	59.0	55.6
Qwen-VLM	64.2	49.4	60.8	62.5	60.2	50.2	61.5	59.2	58.5
MonoQwen	64.5	55.8	**61.8**	**70.6**	68.5	54.0	62.5	**65.7**	62.9
DocReRank-*Base*	**69.6**	**61.5**	60.2	70.1	**69.5**	**56.1**	**64.2**	65.2	**64.6**
DocReRank-*Full*	**67.6**	**64.2**	**62.0**	**72.3**	**69.0**	**59.5**	**65.9**	**67.0**	**66.0**
ColQwen	66.9	58.2	59.6	63.8	66.1	51.3	60.7	60.7	60.9
Qwen-VLM	70.1	54.3	67.1	66.8	61.9	51.3	66.8	59.8	62.3
MonoQwen	71.0	60.3	65.5	**74.8**	70.0	57.1	65.9	**66.6**	66.4
DocReRank-*Base*	**74.9**	**64.0**	**67.2**	**74.8**	**70.8**	**58.6**	**69.1**	66.5	**68.2**
DocReRank-*Full*	**74.3**	**67.0**	**69.5**	**76.7**	**70.7**	**62.6**	**69.5**	**67.3**	**69.7**

Table S3: **Performance on the ViDoReV2 Benchmark NDCG@10**

Benchmark	**FinReport**	**FinSlides**	**TechReport**	**TechSlides**	**Avg**
ColPali	39.6	46.9	67.9	81.7	59.0
Qwen-VLM	45.5	62.1	58.2	63.3	57.3
MonoQwen	**60.8**	67.8	61.1	83.4	68.3
__DocReRank__-Base	58.6	**77.5**	**80.9**	**89.1**	**76.5**
__DocReRank__-Full	**61.4**	**77.6**	**81.2**	**89.4**	**77.4**
ColQwen	44.7	43.2	73.4	84.1	61.4
Qwen-VLM	47.7	58.8	57.6	63.6	56.9
MonoQwen	59.0	63.3	60.7	83.7	66.7
__DocReRank__-Base	**60.5**	**73.7**	**80.6**	**89.5**	**76.1**
__DocReRank__-Full	**64.8**	**73.1**	**81.3**	**89.6**	**77.2**

Table S4: **Performance on the Real-MM-RAG Benchmark recall@1**

Benchmark	**FinReport**	**FinSlides**	**TechReport**	**TechSlides**	**Avg**
ColPali	64.7	76.0	90.1	94.9	81.4
Qwen-VLM	77.1	88.5	86.8	91.5	86.0
MonoQwen	**82.3**	92.6	93.8	**97.7**	91.6
__DocReRank__-Base	81.6	**92.7**	**96.1**	97.5	**92.0**
__DocReRank__-Full	**82.1**	**93.1**	**95.5**	**97.8**	**92.1**
ColQwen	74.9	71.5	93.2	96.5	84.0
Qwen-VLM	80.9	82.5	86.6	91.7	85.4
MonoQwen	89.0	85.7	94.3	98.1	91.8
__DocReRank__-Base	**89.4**	**85.9**	**97.2**	**98.2**	**92.7**
__DocReRank__-Full	**90.0**	**86.5**	**97.1**	**98.4**	**93.0**

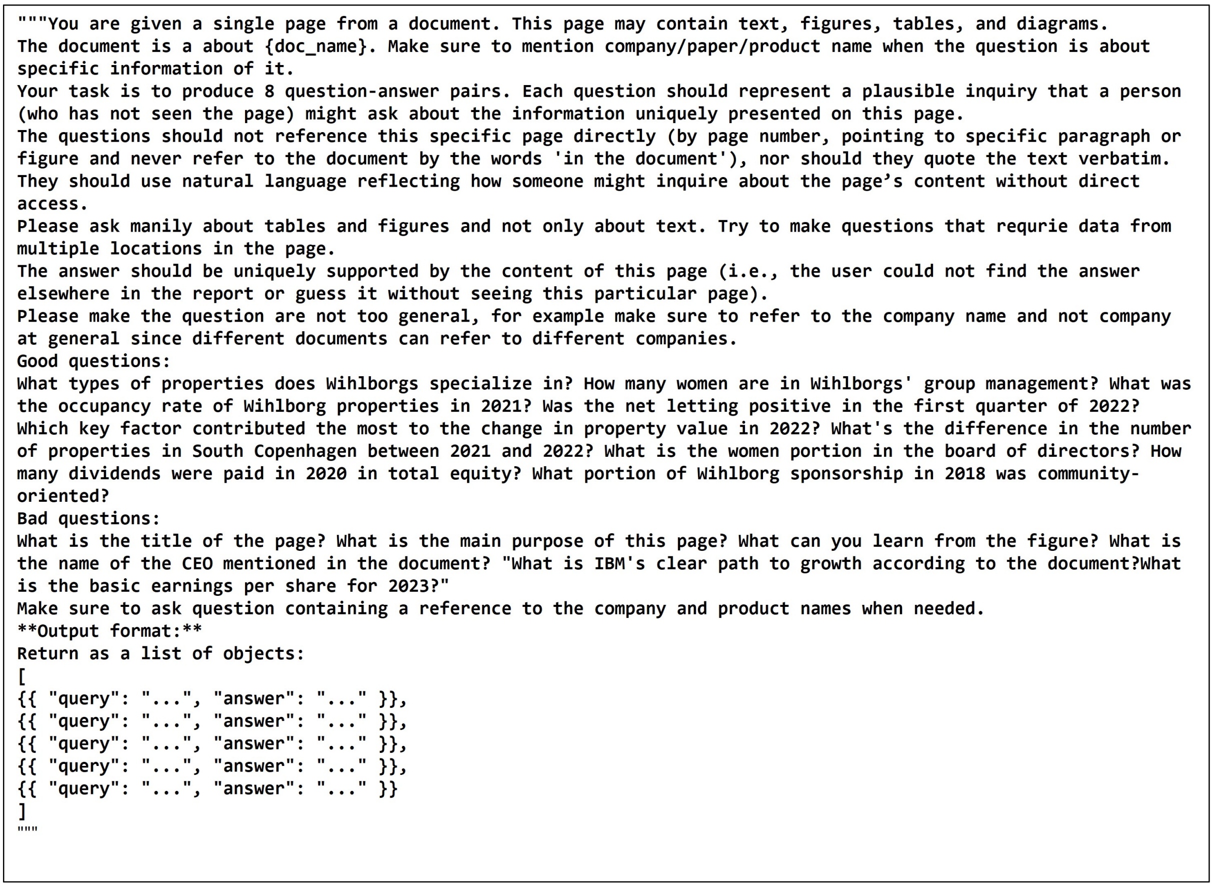
Table S5: **Performance on the Real-MM-RAG Benchmark recall@5**

Benchmark	**FinReport**	**FinSlides**	**TechReport**	**TechSlides**	**Avg**
ColPali	56.0	66.1	81.8	90.0	73.5
Qwen-VLM	64.5	78.2	76.1	80.5	74.8
MonoQwen	**73.7**	82.3	80.3	92.1	82.1
__DocReRank__-Base	72.5	**86.5**	**89.9**	**94.4**	**85.8**
__DocReRank__-Full	**73.8**	**86.9**	**90.0**	**94.5**	**86.3**
ColQwen	64.6	61.6	85.6	91.9	75.9
Qwen-VLM	68.6	72.9	76.0	80.7	74.6
MonoQwen	76.7	76.5	80.6	92.5	81.6
__DocReRank__-Base	**77.8**	**80.8**	**90.4**	**94.9**	**86.0**
__DocReRank__-Full	**79.7**	**80.9**	**90.8**	**95.0**	**86.6**

Table S6: **Performance on the Real-MM-RAG Benchmark NDCG@10**

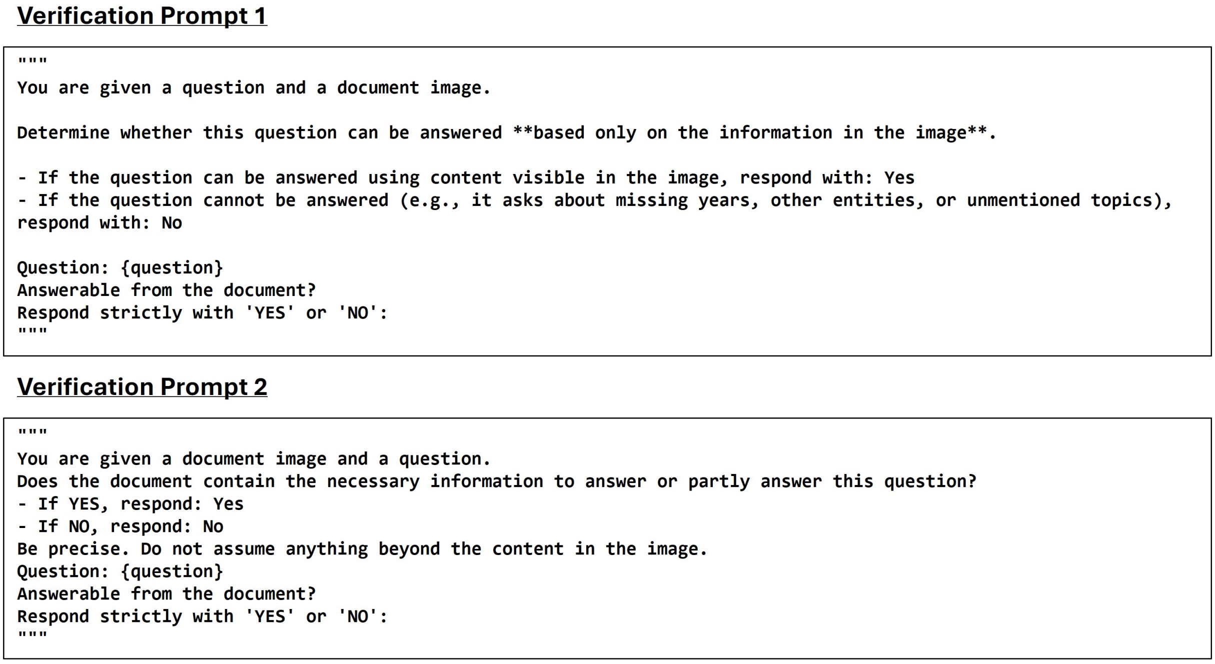
Benchmark	Axa	Economics	Restaurant-rse	Restaurant-esg	Biomedical	Economics-ML	Restaurant-ML	Biomedical-ML	Avg
ColPali	55.0	53.4	51.5	55.2	57.8	47.6	52.5	55.6	53.6
Qwen-VLM	61.8	47.9	57.7	60.8	56.4	49.8	58.0	56.1	56.1
FT on Col-HNDoc	66.4	61.4	59.4	63.3	67.1	55.1	59.2	62.6	61.8
***DocReRank**-Base*	71.8	64.7	58.3	68.8	68.0	58.3	61.2	62.9	64.2
***DocReRank**-B w Fin*	68.5	64.2	**62.2**	68.2	**68.2**	58.3	62.0	63.5	64.4
***DocReRank**-B w Fin&Reph*	68.9	**66.4**	61.6	**69.4**	**68.4**	**60.1**	**62.5**	64.7	**65.3**
***DocReRank**-Full*	**70.7**	67.5	63.4	71.8	66.8	62.8	63.3	**63.7**	66.2

Table S7: **Dataset Ablation Study on ViDoReV2 Benchmark with ColPali**



```
"""You are given a single page from a document. This page may contain text, figures, tables, and diagrams.
The document is a about {doc_name}. Make sure to mention company/paper/product name when the question is about
specific information of it.
Your task is to produce 8 question-answer pairs. Each question should represent a plausible inquiry that a person
(who has not seen the page) might ask about the information uniquely presented on this page.
The questions should not reference this specific page directly (by page number, pointing to specific paragraph or
figure and never refer to the document by the words 'in the document'), nor should they quote the text verbatim.
They should use natural language reflecting how someone might inquire about the page's content without direct
access.
Please ask manily about tables and figures and not only about text. Try to make questions that requrie data from
multiple locations in the page.
The answer should be uniquely supported by the content of this page (i.e., the user could not find the answer
elsewhere in the report or guess it without seeing this particular page).
Please make the question are not too general, for example make sure to refer to the company name and not company
at general since different documents can refer to different companies.
Good questions:
What types of properties does Wihlborgs specialize in? How many women are in Wihlborgs' group management? What was
the occupancy rate of Wihlborg properties in 2021? Was the net letting positive in the first quarter of 2022?
Which key factor contributed the most to the change in property value in 2022? What's the difference in the number
of properties in South Copenhagen between 2021 and 2022? What is the women portion in the board of directors? How
many dividends were paid in 2020 in total equity? What portion of Wihlborg sponsorship in 2018 was community-
oriented?
Bad questions:
What is the title of the page? What is the main purpose of this page? What can you learn from the figure? What is
the name of the CEO mentioned in the document? "What is IBM's clear path to growth according to the document?What
is the basic earnings per share for 2023?"
Make sure to ask question containing a reference to the company and product names when needed.
**Output format:**
Return as a list of objects:
[
{{ "query": "...", "answer": "..." }},
{{ "query": "...", "answer": "..." }},
{{ "query": "...", "answer": "..." }},
{{ "query": "...", "answer": "..." }},
{{ "query": "...", "answer": "..." }}
]
"""
```

Figure S1: **Positive Query Generation Prompt:** Creating RAG style queries, answerable by the corresponding document, using a Pixtral-12B VLM Agrawal et al. (2024). N positive candidates are generated with the given prompt. The prompt emphasizes multimodal understanding by focusing on page elements such as figures, tables, and diagrams.



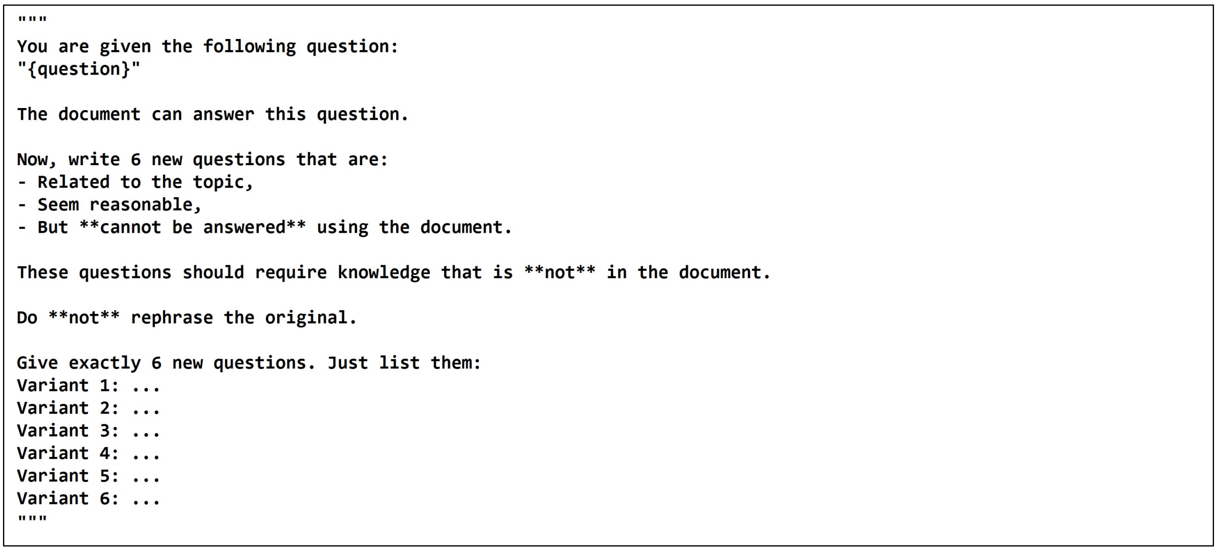
Verification Prompt 1

```
"""
You are given a question and a document image.

Determine whether this question can be answered **based only on the information in the image**.

- If the question can be answered using content visible in the image, respond with: Yes
- If the question cannot be answered (e.g., it asks about missing years, other entities, or unmentioned topics),
respond with: No

Question: {question}
Answerable from the document?
Respond strictly with 'YES' or 'NO':
"""
```

Verification Prompt 2

```
"""
You are given a document image and a question.
Does the document contain the necessary information to answer or partly answer this question?
- If YES, respond: Yes
- If NO, respond: No
Be precise. Do not assume anything beyond the content in the image.
Question: {question}
Answerable from the document?
Respond strictly with 'YES' or 'NO':
"""
```

Figure S2: **Query Verification Prompt.** Verifying whether a query is answerable from the page content using the Qwen2.5-VL-7B-Instruct VLM. Two slightly different prompts are used to improve verification robustness. For positive queries, only those marked as answerable by both prompts are kept; for negatives, any query marked as answerable by either prompt is filtered out.

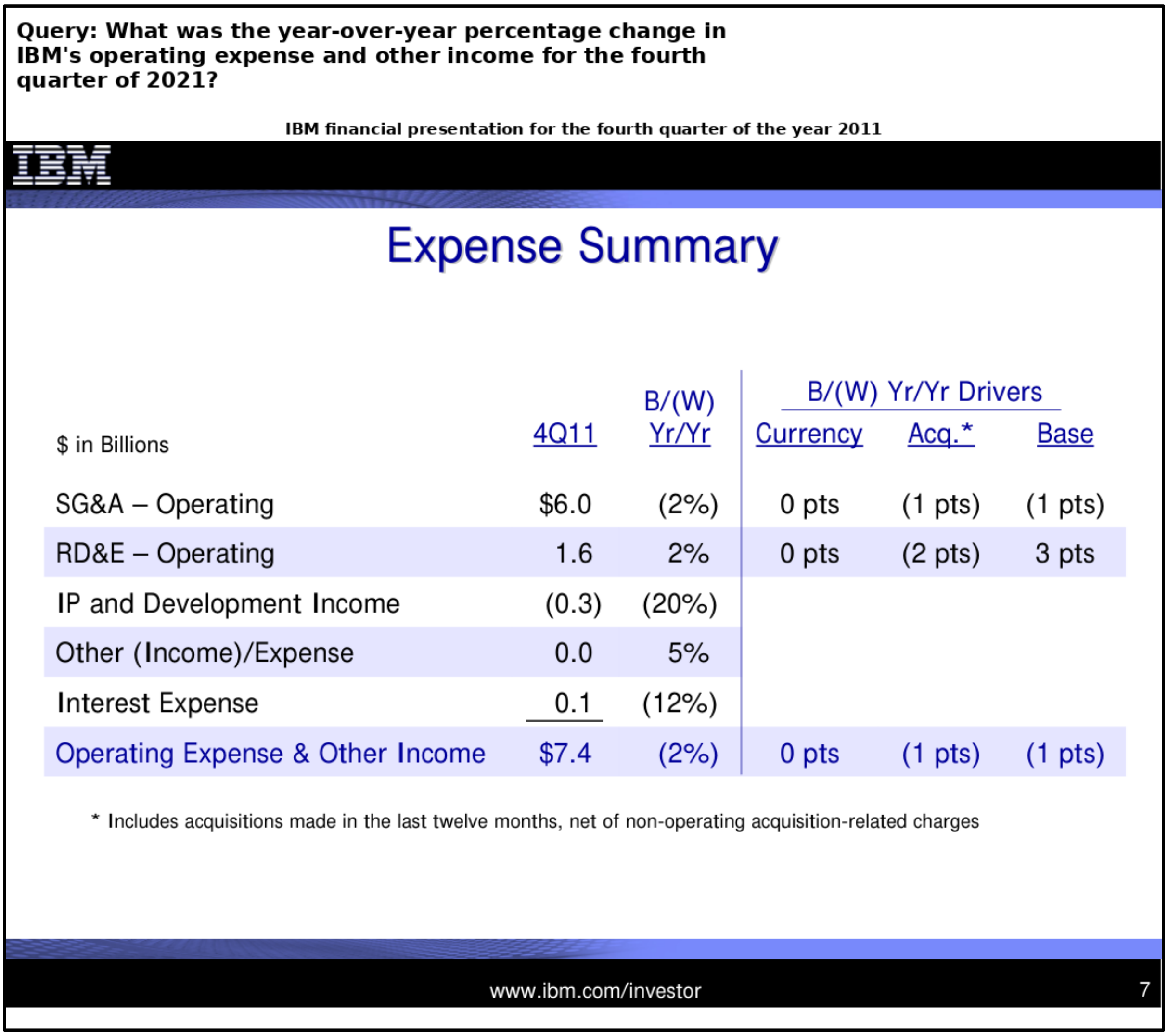


```
"""
You are given the following question:
"{question}"

The document can answer this question.

Now, write 6 new questions that are:
- Related to the topic,
- Seem reasonable,
- But **cannot be answered** using the document.

These questions should require knowledge that is **not** in the document.

Do **not** rephrase the original.

Give exactly 6 new questions. Just list them:
Variant 1: ...
Variant 2: ...
Variant 3: ...
Variant 4: ...
Variant 5: ...
Variant 6: ...
"""
```

Figure S3: **Negative Generation Prompt:** Given a positive query and the corresponding document, hard negative queries - queries that are unanswerable by the current document - are created. As it is relatively easy for an LLM to generate semantically close variants, Qwen2.5-7B-Instruc LLM is used to generate 12 negative variants of the positive query. The prompt is focused to create negatives similar in topic and form, but different in information.

```
"""
We have a valid financial numeric question:
Original question: "{question}"
We want 2 new minimal variants of this question that only change the following property:
{property_desc}
These new variants must NOT be answerable from the same doc,
so they must refer to info not actually in the doc.
Keep the question's sentence structure as close as possible.
Output exactly 2 new questions, labeled:
Variant 1:
Variant 2:
"""
```

Figure S4: **Finance Negative Generation Prompt:** To handle fine-grained information distinctions that occur e.g. in financial reports, a dedicated prompt has been created. A dedicated set of prompts has been developed to generate a variant by modifying exactly one property `{property_desc}`, such as the year (e.g., 2022 → 2024), company name (e.g., Apple → IBM), numerical value (e.g., price, percentage), financial metric (e.g., revenue, sales, acquisitions), subject metric (e.g., dividends, stocks, options), or business segment (e.g., cloud, software, manufacturing).



Query: What was the year-over-year percentage change in IBM's operating expense and other income for the fourth quarter of 2021?

IBM financial presentation for the fourth quarter of the year 2011

IBM

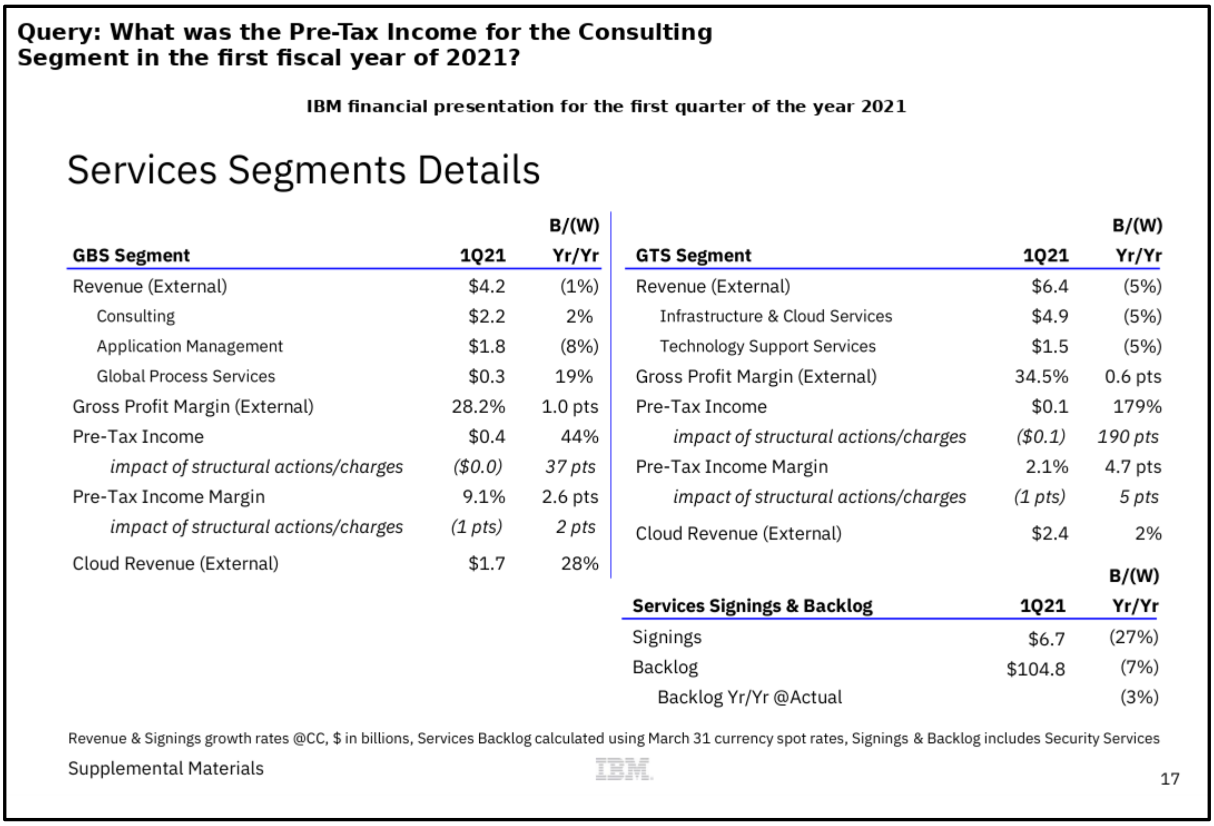
Expense Summary

$ in Billions	4Q11	B/(W) Yr/Yr	B/(W) Yr/Yr Drivers		
			Currency	Acq.*	Base
SG&A – Operating	$6.0	(2%)	0 pts	(1 pts)	(1 pts)
RD&E – Operating	1.6	2%	0 pts	(2 pts)	3 pts
IP and Development Income	(0.3)	(20%)			
Other (Income)/Expense	0.0	5%			
Interest Expense	0.1	(12%)			
Operating Expense & Other Income	$7.4	(2%)	0 pts	(1 pts)	(1 pts)

* Includes acquisitions made in the last twelve months, net of non-operating acquisition-related charges

www.ibm.com/investor 7

Figure S5: **Reranker Failure Case (Wrong Year):** We show an example where the reranker ranked a page first, but it was labeled as negative (i.e., it does not answer the query). This case from FinSlides demonstrates that the model correctly identified relevant cues, such as "expenses," "operating," "year-over-year," and "Q4", but failed on the year: the query asked about 2021, while the retrieved page was from 2011.



Query: What was the Pre-Tax Income for the Consulting Segment in the first fiscal year of 2021?

IBM financial presentation for the first quarter of the year 2021

Services Segments Details

GBS Segment	1Q21	B/(W) Yr/Yr
Revenue (External)	$4.2	(1%)
Consulting	$2.2	2%
Application Management	$1.8	(8%)
Global Process Services	$0.3	19%
Gross Profit Margin (External)	28.2%	1.0 pts
Pre-Tax Income	$0.4	44%
impact of structural actions/charges	*($0.0)*	*37 pts*
Pre-Tax Income Margin	9.1%	2.6 pts
impact of structural actions/charges	*(1 pts)*	*2 pts*
Cloud Revenue (External)	$1.7	28%

GTS Segment	1Q21	B/(W) Yr/Yr
Revenue (External)	$6.4	(5%)
Infrastructure & Cloud Services	$4.9	(5%)
Technology Support Services	$1.5	(5%)
Gross Profit Margin (External)	34.5%	0.6 pts
Pre-Tax Income	$0.1	179%
impact of structural actions/charges	*($0.1)*	*190 pts*
Pre-Tax Income Margin	2.1%	4.7 pts
impact of structural actions/charges	*(1 pts)*	*5 pts*
Cloud Revenue (External)	$2.4	2%

Services Signings & Backlog	1Q21	B/(W) Yr/Yr
Signings	$6.7	(27%)
Backlog	$104.8	(7%)
Backlog Yr/Yr @Actual		(3%)

Revenue & Signings growth rates @CC, $ in billions, Services Backlog calculated using March 31 currency spot rates, Signings & Backlog includes Security Services

Supplemental Materials

17

Figure S6: **Reranker Failure Case (Wrong Business Segment):** We show an example where the reranker ranked a page first, but it was labeled as negative (i.e., it does not answer the query). This case from FinSlides demonstrates that the model correctly identified relevant cues—such as "first quarter 2021," "year-over-year," and "pre-tax income", but failed on the business segment: the query asked about the consulting segment, while the retrieved page referred to the services segment.

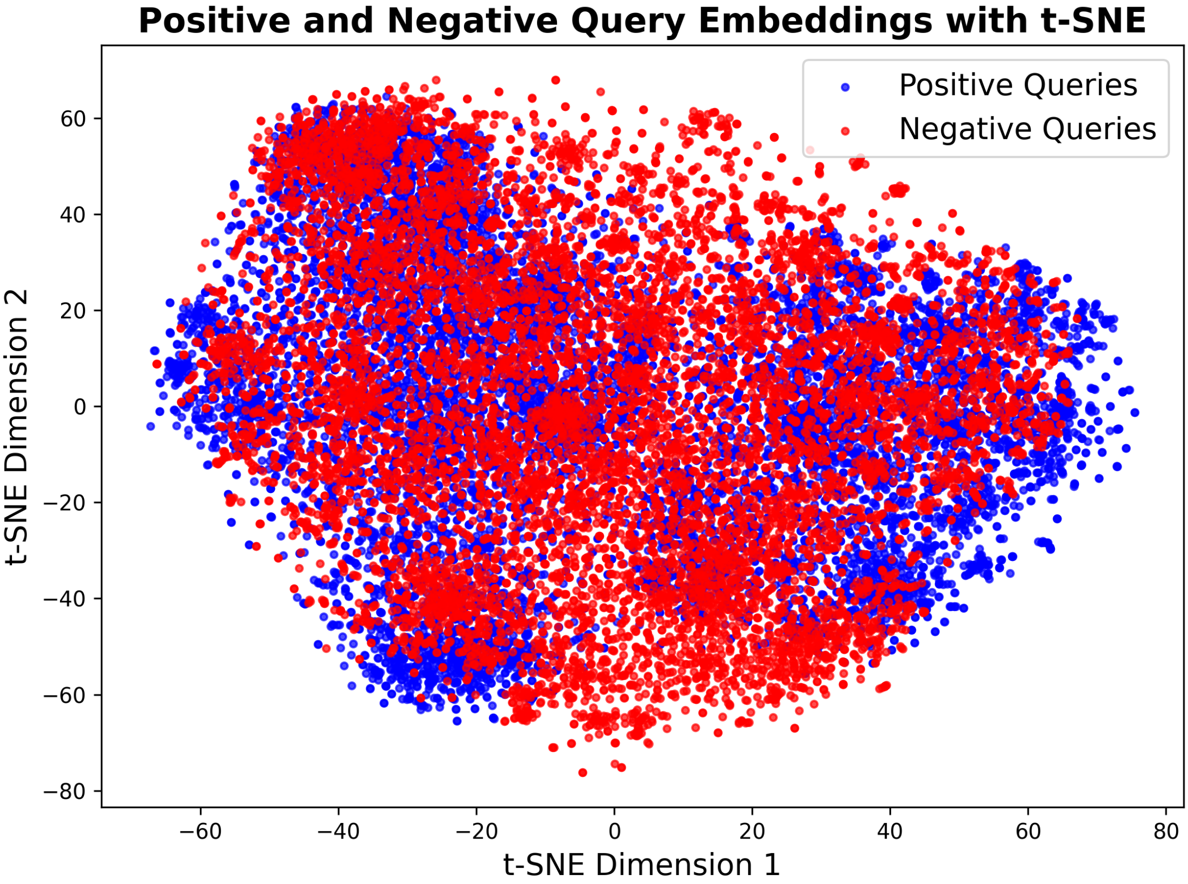


Figure S7: **Visualization of Positive and Negative Query Embeddings with t-SNE:** t-SNE visualization of BGE-M3 (Multi-Granularity, 2024) embeddings for 10,000 randomly sampled positive and generated negative queries from our training set. The results show that the generated negative queries are distributed similarly to the original positive ones.