

# Adaptation of Embedding Models to Financial Filings via LLM Distillation

Eliot Brenner, Dominic Seyler, Manjunath Hegde, Andrei Simion, Koustuv Dasgupta, Bing Xiang  
*Goldman Sachs*

New York, NY, USA

{Eliot.Brenner, Dominic.Seyler, Manjunath.y.Hegde, Andrei.Simion, Koustuv.x.Dasgupta, Bing.Xiang} @gs.com

**Abstract**—Despite advances in generative large language models (LLMs), practical application of specialized conversational AI agents remains constrained by computation costs, latency requirements, and the need for precise domain-specific relevance measures. While existing embedding models address the first two constraints, they underperform on information retrieval in specialized domains like finance. This paper introduces a scalable pipeline that trains specialized models from an unlabeled corpus using a general purpose retrieval embedding model as foundation. Our method yields an average of 27.7% improvement in MRR@5, 44.6% improvement in mean DCG@5 across 14 financial filing types measured over 21,800 query-document pairs, and improved NDCG on 3 of 4 document classes in FinanceBench. We adapt retrieval embeddings (bi-encoder) for RAG, not LLM generators, using LLM-judged relevance to distill domain knowledge into a compact retriever. There are prior works which pair synthetically generated queries with real passages to directly fine-tune the retrieval model. Our pipeline differs from these by introducing interaction between student and teacher models that interleaves retrieval-based mining of hard positive/negative examples from the unlabeled corpus with iterative retraining of the student model’s weights using these examples. Each retrieval iteration uses the refined student model to mine the corpus for progressively harder training examples for the subsequent training iteration. The methodology provides a cost-effective solution to bridging the gap between general-purpose models and specialized domains without requiring labor-intensive human annotation.

**Index Terms**—embedding models, domain adaptation, financial text, retrieval-augmented generation, large language models, knowledge distillation.

## I. INTRODUCTION

Financial services firms are under increasing pressure to extract insights from complex documents like SEC EDGAR filings [18], which pose challenges due to their length, specialized language, and structure [5]. The industry’s adoption of AI demands solutions that accurately retrieve and contextualize this information efficiently.

Retrieval-Augmented Generation (RAG) is a common approach, particularly for conversational AI in finance. Most RAG systems use pipelines that chunk text and transform these chunks into vector representations for fast semantic search. Embedding quality is key, but often lacking in specialized domains like finance [11, 16]. While Large Language Models (LLMs) understand financial context well, their high cost and performance degradation with long context windows necessitates using them with smaller retrieval embedding models.

Our approach leverages an open-weights LLM to generate training data from SEC filings, enabling the fine-tuning of a smaller, more efficient retrieval embedding model that captures nuanced financial semantics. This model improves relevance for financial queries, reduces computational requirements, and accelerates inference. Our approach collects positive and negative samples for fine-tuning a retrieval model over a much larger corpus than previously possible.

Despite advances in long-context models, production financial systems benefit from smaller, specialized models. Long-context models are computationally prohibitive for high-volume, low-latency tasks and have questionable comprehension abilities [15]. Our approach provides domain-specific relevance at a lower computational cost, addressing a critical gap in practical applications.

## II. RELATED WORK

Recent studies highlight the importance of domain-specific models in finance. Peng et al. [13] found that models continuously pretrained on financial text outperform those trained on general text, while Tang and Yang [16] demonstrated performance drops when general embedding models are applied to finance-specific tasks. Furthermore, Li et al. [11] showed that general models often struggle with specialized vocabulary and document structures in financial filings. Wang et al. [19] found the choice of retrieval embedding model significantly impacts downstream generator performance in financial contexts. These findings emphasize the need for specialized financial models.

Anderson et al. [3] introduced a closely related work, presenting text embeddings fine-tuned on 14.3 million financial query-passage pairs. They demonstrated improved retrieval embedding models via domain-specific training, achieving a 62.8% Recall@1 versus 39.2% for OpenAI’s best general-purpose embeddings, highlighting the challenges of financial text. Our method differs primarily in its mining of positive examples, using an LLM judge to generate roughly  $10^3$  positive passages per query (over two iterations), compared to their generally single positive passage per query. We also mine hard negative examples from within the same documents as the positive examples, using LLM relevance judgments and considering all retrieval ranks, whereas they source negatives from fixed ranks within the entire corpus without judging irrelevance. This allows us to make the model better adapted for within-document retrieval. Our approach generates queries

TABLE I: Statistics of the Training Corpus

Category	Type	# Docs (K)	Avg. Chunks	# Chunks (M)
Financial Report	10-K	67	123	8.23
	10-Q	36	85	3.10
	6-K	28	56	1.55
	8-K	79	72	5.72
Prospectus	424B3	5	440	2.55
	424B4	0.2	1,664	0.38
	497K	12	78	0.92
Amendment	485BPOS	17	499	8.94
	SC 13D/A	7	44	0.3
	SC 13G/A	22	20	0.44
Pricing Details	424B2	101	167	16.87
Proxy Statement	DEF 14A	4	598	2.56
Registration	S-8	8	24	0.18
Ownership Report	SC 13G	8	19	0.15
<b>Total:</b>		<b>396</b>		<b>51.88</b>

by few-shot prompting which encourages the generation of queries which are not as closely tied to particular passages, incorporates document class and query association with a single document class, and enables the calculation of IR metrics like mean DCG@ $k$  and MRR@ $k$  due to LLM-based reward step labeling of *each* of the top- $k$  chunks in each document (i.e., retrieval set).

In the finance domain, Zhao et al. [21] systematically benchmark RAG pipeline components on a real financial dataset, underscoring the importance of retrieval quality and domain-tailored evaluation.

Because SEC reports are highly standardized and long, general RAG pipelines under-retrieve or duplicate boilerplate. Concurrent work on filings-native RAG Choe et al. [6] addresses near-duplicate sections in filings at scale, and DocFinQA Reddy et al. [14] shows retrieval is essential on *full filings*. Our work complements these by adapting the retriever itself via LLM-judged distillation to improve within-document passage selection across 14 filing types.

### III. DATASET

We collected a dataset by crawling SEC filings documents, amounting to 396,165 documents distributed over 14 filing types, from the SEC website [18], published in the time frame of January 1st, 2024 to July 31st, 2024.

We divide our dataset as follows: January 1st through June 30th data are used for training and validation purposes, whereas the month of July data are held-out for testing.

Table I shows the statistics of the training corpus. See [17] for a description of each filing type.

### IV. METHOD

We give a conceptual overview of the main pipeline in Section IV-A (Figure 1 shows the main pipeline and Figure 2 the evaluation pipeline) and then explain certain details and modifications in Sections IV-B-IV-D.

#### A. Scaffold Iterative Process

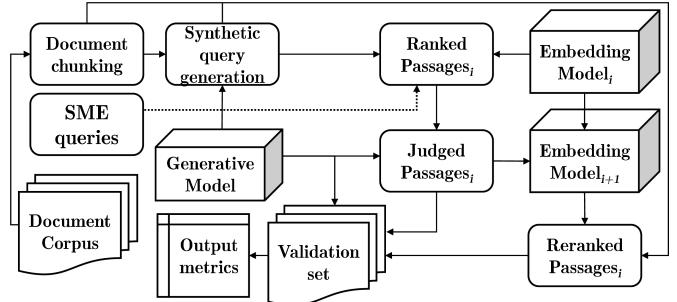


Fig. 1: Training/validation pipeline overview. The iterative components of the pipeline are subscripted with “ $i$ ”.

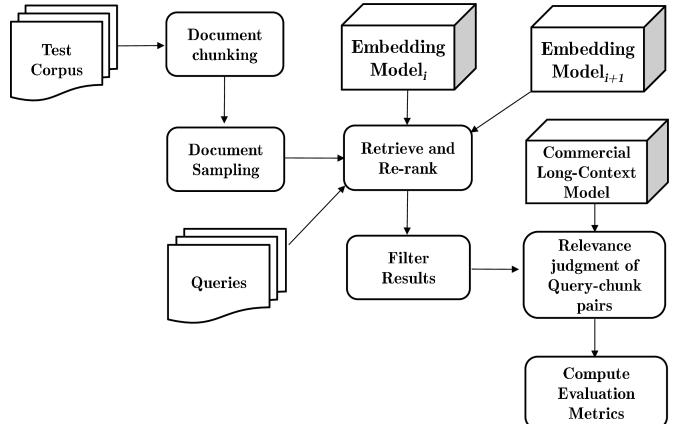


Fig. 2: Final evaluation pipeline. The iterative components of the pipeline are subscripted with “ $i$ ”. The test corpus is fixed for all iterations.

**Prior to all iterations:** The corpus  $D$  of documents  $d \in D$  is divided into the disjoint union of  $D_{\text{train-val}}$  (for brevity,  $D_{\text{tv}}$ ) and  $D_{\text{test}}$ .  $D_{\text{tv}}$  is similarly sub-divided into the disjoint union of  $D_{\text{train}}$  and  $D_{\text{val}}$ . Thus, for example

$$D_{\text{tv}} = D_{\text{train}} \cup D_{\text{val}}. \quad (1)$$

The corpus is chunked into chunks of between 500 to 1000 characters, respecting sentence boundaries where possible. Call the resulting corpus of chunks  $C$ , and note that for our corpus,  $|C| \approx 5 \times 10^7$ . The chunk set  $C$  inherits the splitting from  $D$  into  $C_{\text{tv}}$ ,  $C_{\text{test}}$ ,  $C_{\text{train}}$  and  $C_{\text{val}}$ . Denote by  $C_d$  the chunks of any fixed  $d \in D$ . Thus,

$$C = C_{\text{tv}} \cup C_{\text{test}},$$

$$C_{\text{tv}} = C_{\text{train}} \cup C_{\text{val}},$$

$$C = \bigcup_{d \in D} C_d,$$

with all the above unions being disjoint.

**In the  $i$ th iteration ( $i \geq 0$ ):** Denote by  $\text{bi-enc}(i)$  the state of the bi-encoder model at the start of iteration  $i$ . Set  $\text{bi-enc}(0)$  equal to the gte-large model (see Section V below), and for

$i > 0$ , set  $\text{bi-enc}(i)$  equal to the fine-tuned version created in iteration  $i - 1$ .

**Step 1.** Provide the teacher LLM with (a sample) of  $c \in C_{\text{tv}}$  and prompt it to generate queries relevant to each  $c$ . For generating queries using an LLM we use methods based on InPars [4], which means that we prompt the LLM for a large number of queries and select the top-scoring queries according to conditional log probability (see below for more details). We compared the “vanilla” and “gbq” prompt versions but opted to use the vanilla version in our work. For each document we sample 500 chunks and generate one question per chunk. Following [4], we select the top  $k = 200$  chunks with the highest token probability. After this selection, the selected query set is denoted  $Q_i$ .

**Step 2.** For each  $q \in Q_i$ , create a *sample*  $D_q \subset D_{\text{tv}}$ , inheriting a splitting  $D_q = D_{q,t} \cup D_{q,v}$  from the splitting (1). Define  $C_{q,t} := \bigcup_{d \in D_{q,t}} C_d$ ,  $C_{q,v} := \bigcup_{d \in D_{q,v}} C_d$ , and  $C_q := C_{q,t} \cup C_{q,v}$ . (disjoint union)

**Step 3.** Prompt the teacher LLM to assign an ordinal relevance score  $r_{q,c} \in [1, 4] \cap \mathbb{Z}$  to a *sample* of the  $c \in C_q$ . The criteria provided to the LLM in scoring prompt are given in Table II. The collection of triples  $(q, c, r_{q,c})$  so obtained serves as a raw training set for the subsequent steps.

TABLE II: LLM Relevance Scoring Criteria for Passage-Query Matching

Score	Criteria in LLM Prompt
1	No answer and not relevant - The query is entirely unrelated to the content of the passage and contains no answer.
2	No answer but somewhat relevant - The query has weak relevance to the content of the passage, the connection between them is unclear or incomplete, we can not answer the query based on the passage.
3	Partial answer and moderately relevant - The query is relevant to the content of the answer passage and shows a clear connection, but there may be some gaps or minor inconsistencies in relevance. If we want to answer the query accurately based on the passage as context, the answer will be partial.
4	Explicit answer and perfectly relevant - The passage is perfectly and directly relevant to the query, with a clear and complete connection between them. The answer is not only relevant but also highly accurate, effectively and explicitly answering the query.

**Step 4.** Extract from the triples  $(q, c, r_{q,c})$  contrastive triples  $(q, c, c')$  satisfying  $r_{q,c} > r_{q,c'}$ . We enforce the following constraints:

- We only use positive and negative examples from the same document to form a triple, so  $d_c = d_{c'}$ .
- We only consider  $(q, c)$  as positive if  $r_{q,c} > 3$ . and negative if  $r_{q,c} < 3$ .

After applying these constraints and filtering out duplicates we obtain 2.52 million triples for training and 950K triples for validation.

Denote the triples so obtained as  $(q, c_{\text{rel}}, c_{\text{irrel}})$ , the notation indicating that  $c_{\text{rel}}$  is strictly more relevant to  $q$  than  $c_{\text{irrel}}$ . We emphasize that *in all the triples, the queries are synthetic but the passages are real chunks from the corpus*.

**Step 5. Student model retraining.** Use the triples concatenated from Step 4 in iterations  $0, \dots, i$ , and triplet loss  $\mathcal{L}_{\text{triplet}}$  to

retrain the student model.

$$\begin{aligned} \mathcal{L}_{\text{triplet}} := \max & \left( 0, \alpha + d(\text{bi-enc}(q), \text{bi-enc}(c_{\text{rel}})) \right. \\ & \left. - d(\text{bi-enc}(q), \text{bi-enc}(c_{\text{irrel}})) \right) \end{aligned}$$

Our encoder model was fine-tuned on the triples dataset discussed prior. We choose the following hyperparameters to train the model: Number of training epochs: 2, learning rate:  $5 \times 10^{-7}$ , batch size: 128 (16 per GPU), Optimizer: Adam-W. After training the model triples accuracy on the validation set increased from  $\sim 95\%$  to  $\sim 98\%$  for experimental and the control variant. Further, to counteract catastrophic forgetting and stabilize the model’s performance, we needed to include a large number of triples that were already correctly ranked by the base model. In addition, the model was highly sensitive to the margin parameter, which we found the best setting to be 0.1 (compared to 5, which is the default setting in the sentence transformer library).

If the model prior to retraining is denoted  $\text{bi-enc}(i)$ , then denote the model after retraining by  $\text{bi-enc}(i+1)$ .

**Step 6. Evaluation for Hyperparameter Tuning:** Using the relevance judgments  $r_{q,c}$  for  $c \in C_{q,v}$  and  $q \in Q(i)$ , calculate IR metrics (e.g., DCG) for  $\text{bi-enc}(i)$  and  $\text{bi-enc}(i+1)$ .

### B. Address Sparsity of Answers in Corpus

1) *Associate Document Class to Queries:* Given that each filing type serves a distinct reporting role, the corpus is treated as 14 sub-corpora, for the purposes of query-generation and search. To create a unified embedding model for all filing types, the methodology adapted is as follows:

Each document class (filing type) is associated with specific queries relevant to it. Training data generation (Steps 1-4) and Evaluation on Validation Set (Step 6) are performed independently for each filing type. Data from all filing types is mixed (concatenated) only in Step 5, bi-encoder model retraining. Thus, each generated query is linked to a specific filing type.

2) *Corpus-wide Retrieval to Select  $C_{q,t}$ :* To address the sparsity of answers to a query  $q$  within the documents  $D$  of the filing type associated to  $q$ , in order to form  $D_q$  we perform retrieval using  $\text{bi-enc}(i)$ , and  $q$  as anchor text, over all of  $C_{\text{tv}}$ . Set

$$\begin{aligned} C_{\text{tv}}(q, K, i) := & \text{top } - K \text{ closest } c \in C_{t,v} \\ & \text{to } q \text{ under } \text{bi-enc}(i). \end{aligned}$$

Then set

$$D_{\text{tv}}(q, K, i) := \{d \in D_{\text{tv}} \mid C(d) \cap C_{\text{tv}}(q, K, i) \neq \emptyset\},$$

the idea being that defining  $D_q$  as  $D_{\text{tv}}(q, K, i)$  makes it more likely that a higher proportion of  $d \in D_q$  contain an answer to  $q$  than documents chosen at random. Note that as a consequence of these definitions and the characteristics of our corpus (and similar corpora)  $|C(D_{\text{tv}}(q, K, i))|$  is typically as large as  $10^3 \times K \approx 10^5$ . This observation motivates our next set of adjustments to the core method.

TABLE III: Selected Notation

Notation	Definition
$D$	Corpus of documents $d$
fold	Generic way of referring to train, validation (val), test, or train-validation (t-v) folds
$D_{\text{fold}}$	Documents in fold
$C$	Chunked corpus
$C_d$	Chunks of $d \in D$
$C_{\text{fold}}$	$\cup_{d \in D_{\text{fold}}} C_d$
$Q(i)$	Queries generated in iteration $i$ , $i \geq 0$
$D_{q,\text{fold}}$	A sample of $d \in D_{\text{fold}}$ most likely to have answers to $q \in Q(i)$
$r_{q,c} \in [1, 4] \cap \mathbb{Z}$	Relevance of $c$ to $q$ as assigned by LLM
$(q, c_{\text{rel}}, c_{\text{irrel}})$	Training triple satisfying $r_{q,c_{\text{rel}}} > r_{q,c_{\text{irrel}}}$
$\text{bi-enc}(i)$	Student model in iteration $i$ , baseline model for $i = 0$ and model trained in iteration $i - 1$ for $i > 0$
$\text{metric}_q(i)$	Metric of $q$ calculated on $c \in C_{q,v}$ using $\text{bi-enc}(i)$ as retriever
$K$	Positive integer parameter determining the maximum number of $d \in D_{\text{tv}}$ considered in-scope for all $q \in Q(i)$
$k$	Positive integer parameter reflecting the number of passages to be used as context in RAG.
$C_{\text{fold}}(q, K, i)$	Top- $k$ closest $c \in C_{\text{fold}}$ to $q \in Q(i)$ under $\text{bi-enc}(i)$
$D_{\text{fold}}(q, K, i)$	$\{d \in D_{\text{fold}} \mid C(d) \cap C_{\text{fold}}(q, K, i) \neq \emptyset\}$
$C_d(q, k, i)$	top- $k$ closest $c \in C(d)$ under $\text{bi-enc}(i)$
$C_d(q, k)$	Union of $C_d(q, k, i)$ and $2k$ randomly sampled $c \in C_d \setminus C_d(q, k, i)$
$P$	Integer parameter such that the top $1/P$ fraction of $(q, d) \in \cup_{q \in Q(i)} \{q\} \times D_{\text{v}}(q, i)$ are evaluated
$M$	Integer parameter determining the number of total distinct queries obtained for each document class as $2M$ .
$C_{\text{test}}(q, K, i)$	top- $K$ closest $c \in C_{\text{test}}$ to $q$ under $\text{bi-enc}(i)$ , for $q \in Q$ .
$C_{\text{test}}(q, K)$	$\bigcup_{i=0, \dots, N} C_{\text{test}}(q, K, i)$
$D_{\text{test}}(q, K)$	$\{d \in D_{\text{test}} \mid C(d) \cap C_{\text{test}}(q, K) \neq \emptyset\}$
$C_{\text{test}}(q, d, i, k)$	top- $k$ $c \in C(d)$ closest to $q$ under $\text{bi-enc}(i)$

### C. Improve Scalability to Large Corpora

1) *Random Sampling and Corpus Partition:* To decrease the computational cost of retrieving  $C(D_{\text{tv}}(q, K, i))$ , at the beginning of iteration  $i$  of the pipeline, a sample  $D(i)$  is drawn from  $D_{\text{tv}}$  to serve in place of  $D_{\text{tv}}$  in iteration  $i$ . To ensure a more equitable distribution over document classes a number of documents is sampled per class, determined so as to achieve  $|C(D)| \approx 10^6$  per class. Similarly, to account for the varying distribution of filing type dates over the time window, the train/validation split is performed by choosing a *filing-type dependent* split date in order to achieve a roughly 70%/30% training/validation triples split for each filing type.

2) *Downsampling within  $C(D_{\text{tv}}(q, K, i))$ :* Judging every query-chunk pair would give on the order of  $10^9$  calls to the LLM, and to cut down on this number,  $C(D_{\text{tv}}(q, K, i))$  is further downsampled. We posit that, in spite of any deficiencies in  $\text{bi-enc}(i)$ , relevance of chunks in  $d \in D_{\text{tv}}(q, K, i)$  to  $q$  still correlates heavily with their similarity to  $q$  under  $\text{bi-enc}(i)$ . Therefore, we sample chunks to actually send to the LLM for judgment in the following way:

**Step 3a** For each  $d \in D_{\text{tv}}(q, K, i)$  rank the  $c \in C_d$  using

distance under  $\text{bi-enc}(i)$ . Form  $C_d(q, i)$  by taking the union of

$$C_d(q, k, i) := \text{top } - k \text{ closest } c \in C_d \text{ under } \text{bi-enc}(i)$$

and of a random sample of cardinality  $2k$  is drawn from

$$C_d \setminus C_d(q, k, i),$$

using a parametric, non-uniform probability distribution over the ranks  $r = k, \dots, |C(d)|$  (*ranks* means the ordinals assigned to chunks by scoring each chunk relative to  $q$  with  $\text{bi-enc}(i)$  as the embedding model). The probability distribution over ranks is defined as follows: assign to each rank  $r$ , the *unnormalized weight*

$$w_i = \exp^{-\omega(r-k)}, \quad \omega \text{ a constant.}$$

Then, normalize the  $w_i$  to sum to 1 and thereby obtain a probability distribution. In this way, we achieve the aim of making higher ranks (exponentially) more likely to be sampled than lower ranks. Note that by definition

$$|C_d(q, i)| = k + 2k = 3k \ll |C_d|.$$

For it to be possible to calculate MRR@ $k$  and similar metrics at  $k$  in Step 6, we need to have judgments relevance  $C_d(q, k, j)$  not only for  $j = i$ , but also for  $j = i + 1$ . According to **Step 3a** only the former, not the latter, are obtained prior to model fine-tuning. Therefore, we have to add an additional step:

**Step 6a.** For each  $(q, c) \in C_d(q, k, i + 1)$  (for each  $d \in D_{\text{v}}(q, i)$  only) prompt the *judging* generative model to assign relevance score  $r(q, c)$ .

This enables the calculation of MRR@ $k$  and DCG@ $k$ , but not beyond  $k$ , meaning that the hyperparameter  $k$  must be chosen with care at initialization, taking into consideration to the intended use of the retriever. We choose  $k = 5$ . Note that prior to computation of all metrics we used a threshold ( $\geq 4$ ) to binarize the relevance score  $r(q, c)$  into  $\{0, 1\}$ .

### D. Generation of “Synthetic” Queries

To address InPars’ tendency to generate chunk-specific queries, we generate “synthetic” queries using a method which is *not* conditioned on a specific chunk. Starting with  $M$  InPars-generated queries, we augment this set with synthetic queries using few-shot prompting:

**Step 1a. Generation of synthetic queries via few-shot prompting.** Initialize set of synthetic queries as  $\emptyset$ . Until there are  $M = 200$  synthetic queries repeat the following: select 5 InPars queries (“exemplars”); prompt LLM( $i$ ) to *generate, as an SME, 5 additional queries based on the 5 exemplars*; after rejecting any queries which are too short to be questions or fail other simple heuristic quality measures, add the generated queries to the synthetic queries set. Together with the InPars queries this results in  $14 * 400 = 5600$  total queries.

To encourage query diversity we sample exemplars from distinct clusters, clusters being based on the embedding model  $\text{bi-enc}(0)$ .

**Listing 1: LLM Question Generation Prompt**

```
You are a seasoned financial expert meticulously
reviewing earnings call transcripts with a laser
focus, and expertly crafting insightful
questions to distill the critical insights
trends within.
Your task is to generate a set of queries that
financial experts would ask about an earning
call transcript.

# Example of queries:
{list of existing queries}

generate a set of five new queries, you can
familiarize yourself with the nature of queries
using the data above. each query should be
generated in a new line.
```

**E. Final Evaluation Pipeline**

A separate final evaluation pipeline is needed to address biases present when using only the main training/validation pipeline. The main pipeline compares bi-enc( $i$ ) and bi-enc( $i+1$ ) solely on  $Q_i$ , potentially biasing towards bi-enc( $i+1$ ). The final evaluation pipeline generates new, out-of-sample queries,  $Q_{\text{test}}$ , using the InPars method with sampled passages  $d$  from a held-out dataset  $D_{\text{test}}$ . This allows for measuring retrieval performance on  $D_{\text{test}}$  instead of the training/validation set  $D_{tv}$ . Furthermore, to mitigate biases from the open-weights teacher model, a different (commercial) LLM (GPT-4o) is employed as judge in the final evaluation pipeline.

**1) Scaffolding of Evaluation Pipeline:** **Step 1.** Denote by  $N$  the number of iterations of the main training pipeline. For each  $q \in Q_{\text{test}}$ , set

$$C_{\text{test}}(q, K, i) := \text{top } - K \text{ closest } c \in C_{\text{test}} \text{ to } q \text{ under bi-enc}(i),$$

$$C_{\text{test}}(q, K) := \bigcup_{i=0, \dots, N} C_{\text{test}}(q, K, i).$$

Define

$$D_{\text{test}}(q, K) := \{d \in D_{\text{test}} \mid C(d) \cap C_{\text{test}}(q, K) \neq \emptyset\}.$$

Each of Steps 2 and 3 below is repeated for  $i$  ranging over  $0, \dots, N-1$ .

**Step 2.** For each  $d \in D_{\text{test}}(q, K)$ , define

$$C_{\text{test}}(q, d, i, k) := \text{top } - k \text{ } c \in C(d) \text{ closest to } q \text{ under bi-enc}(i). \quad (2)$$

**Step 3.** For fixed  $i \in \{0, \dots, N-1\}$ ,  $j$  is a variable taking values in  $\{i, i+1\}$ . Based on certain characteristics of  $C_{\text{test}}(q, d, j, k)$  (to be described below), score all  $d \in D_{\text{test}}(q, K)$  and define  $D_{\text{test}}^{(i)}(q, K) \subset D_{\text{test}}(q, K)$ , to be the top-scoring  $d$ , so that  $|D_{\text{test}}(q, K)| / |D_{\text{test}}^{(i)}(q, K)| \approx P$ , for  $P \in \mathbf{Z}$  fixed hyperparameter. Define  $C_{\text{test}}^{(i)}(q, d, i, k)$  by (2) for  $d \in D_{\text{test}}^{(i)}$  and as  $\emptyset$  for  $d \in D_{\text{test}} - D_{\text{test}}^{(i)}$ . Repeat the following until the confidence intervals of all metrics are small: randomly sample  $d \in D_{\text{test}}^{(i)}(q, K)$ ; use the LLM to assign  $r(q, c)$  to  $(q, c) \in C_{\text{test}}^{(i)}(q, d, j, k)$ ; recompute metrics and their confidence intervals @ $k$  based on all  $r(q, c)$  assigned so far.

In lieu of completely specifying the score in Step 3, we highlight that the following key point: the score depends directly on two factors which can be computed for any  $d \in D_{\text{test}}(q, K)$ :

- 1) The following (modified Hausdorff) *distance*:  $\text{dist}(C_{\text{test}}(q, d, i, k), C_{\text{test}}(q, d, i+1, k))$ .
- 2) The following *similarity*, for  $S$  a constant:  $\text{relu}(S - \min_{j=i, i+1} \text{dist}(q, C_{\text{test}}(q, d, j, 1)))$ .

These factors embody the propensity of  $d$  to contain text relevant to  $q$  and the supervised fine tuning in iteration  $i$  to change the ranking of  $C(d)$  in a meaningful manner. The stopping criterion for the evaluation in **Step 3** which we used is that, for each metric, the standard error is  $< 5\%$  of the estimated mean. The variation in  $D_{\text{test}}^{(i)}$  and of the  $(q, c)$  actually evaluated, explains why in Table IV, we have to report “counts” of  $(q, c)$  underlying the reported metrics and also why the experiment metrics for iteration 0 do not match the base metrics for iteration 1, although the model underlying these metrics is the same.

## V. MODEL SELECTION

As our embedding model of choice we use the large variant of the General Text Embeddings (GTE) model [10]. It was chosen due to it being a widely-used model optimized for information retrieval downstream tasks. Further, its size of 335 million parameters makes it highly scalable and easy to use in a production setting. We prefer GTE-large over the smaller MiniLM model [20] due to GTE’s superior performance [10]. As our generative LLM we chose the Llama-3.1-70B-Instruct [1] as it presented the best performance among open-weight models of that size [2]. We chose the 70B parameter version of the model, as we could not achieve an adequately large throughput on our hardware (8x Nvidia H100) using a larger variant of the model.

## VI. RESULTS

### A. Quantitative Results

Table IV reports performance metrics, calculated as described in Section IV-E for different filing types across two training iterations. Table V and Figure 3 report performance metrics of bi-enc( $i$ ) for  $i = 0, 1, 2$  on the public evaluation set of FinanceBench [9]. Although we did not have access to the FinanceBench training set, we performed this “out-of-distribution” evaluation to enhance reproducibility and provide a stringent challenge for our methods. We (deterministically) obtained relevance labels  $r(q, c)$  for all  $c \in C(d)$  using the “evidence passages” present in FinanceBench, as explained below in Section VI-B. In all reporting tables, the metric followed by “d” indicates the Cohen’s  $d$  [7], a widely used statistical measure of effect size. The Cohen’s  $d$  values indicate that the training of the base embedding model in the first iteration made a statistically significant difference in ranking performance. In the second training iteration, although most metrics remain positive or show improvement in the measured values compared to their baselines, the changes are smaller.

TABLE IV: Result for final evaluation pipeline on all filing types. In “Iteration  $i$ ”, the “Base”, resp. “Exp” version of the metric@5 is calculated with bi-enc $i$  (resp. bi-enc( $i + 1$ )) as the retriever.  $\mu$  denotes the average.

Type	Count	Iteration 0			Iteration 1			
		Base	Exp	Cohen’s d	Count	Base	Exp	Cohen’s d
485BPOS	1430	MRR: 0.14 $\mu$ DCG: 0.2	MRR: 0.18 $\mu$ DCG: 0.27	MRR: 0.099 $\mu$ DCG: 0.13	1430	MRR: 0.18 $\mu$ DCG: 0.27	MRR: 0.18 $\mu$ DCG: 0.27	MRR: 0.0048 $\mu$ DCG: 0.005
6-K	1801	MRR: 0.11 $\mu$ DCG: 0.14	MRR: 0.16 $\mu$ DCG: 0.21	MRR: 0.15 $\mu$ DCG: 0.17	1801	MRR: 0.18 $\mu$ DCG: 0.24	MRR: 0.19 $\mu$ DCG: 0.25	MRR: 0.015 $\mu$ DCG: 0.025
SC 13D/A	1799	MRR: 0.091 $\mu$ DCG: 0.13	MRR: 0.13 $\mu$ DCG: 0.21	MRR: 0.14 $\mu$ DCG: 0.17	1381	MRR: 0.14 $\mu$ DCG: 0.2	MRR: 0.13 $\mu$ DCG: 0.2	MRR: -0.0042 $\mu$ DCG: 0.0077
8-K	2460	MRR: 0.13 $\mu$ DCG: 0.16	MRR: 0.16 $\mu$ DCG: 0.21	MRR: 0.09 $\mu$ DCG: 0.12	1858	MRR: 0.17 $\mu$ DCG: 0.22	MRR: 0.18 $\mu$ DCG: 0.23	MRR: 0.022 $\mu$ DCG: 0.029
424B4	369	MRR: 0.1 $\mu$ DCG: 0.14	MRR: 0.14 $\mu$ DCG: 0.22	MRR: 0.13 $\mu$ DCG: 0.16	369	MRR: 0.15 $\mu$ DCG: 0.24	MRR: 0.16 $\mu$ DCG: 0.25	MRR: 0.0087 $\mu$ DCG: 0.028
SC 13G/A	1719	MRR: 0.17 $\mu$ DCG: 0.29	MRR: 0.27 $\mu$ DCG: 0.56	MRR: 0.26 $\mu$ DCG: 0.34	1013	MRR: 0.32 $\mu$ DCG: 0.6	MRR: 0.32 $\mu$ DCG: 0.61	MRR: -0.012 $\mu$ DCG: 0.00083
DEF 14A	1249	MRR: 0.23 $\mu$ DCG: 0.27	MRR: 0.3 $\mu$ DCG: 0.41	MRR: 0.17 $\mu$ DCG: 0.25	769	MRR: 0.36 $\mu$ DCG: 0.48	MRR: 0.36 $\mu$ DCG: 0.5	MRR: -0.002 $\mu$ DCG: 0.017
424B3	1108	MRR: 0.11 $\mu$ DCG: 0.16	MRR: 0.13 $\mu$ DCG: 0.22	MRR: 0.072 $\mu$ DCG: 0.12	1108	MRR: 0.16 $\mu$ DCG: 0.26	MRR: 0.16 $\mu$ DCG: 0.25	MRR: -0.012 $\mu$ DCG: -0.0064
10-K	1349	MRR: 0.13 $\mu$ DCG: 0.16	MRR: 0.15 $\mu$ DCG: 0.21	MRR: 0.063 $\mu$ DCG: 0.11	1349	MRR: 0.18 $\mu$ DCG: 0.25	MRR: 0.19 $\mu$ DCG: 0.26	MRR: 0.015 $\mu$ DCG: 0.025
497K	1759	MRR: 0.19 $\mu$ DCG: 0.23	MRR: 0.2 $\mu$ DCG: 0.25	MRR: 0.029 $\mu$ DCG: 0.047	1467	MRR: 0.23 $\mu$ DCG: 0.29	MRR: 0.23 $\mu$ DCG: 0.3	MRR: 0.0079 $\mu$ DCG: 0.021
424B2	1214	MRR: 0.25 $\mu$ DCG: 0.36	MRR: 0.31 $\mu$ DCG: 0.47	MRR: 0.12 $\mu$ DCG: 0.16	1008	MRR: 0.31 $\mu$ DCG: 0.47	MRR: 0.31 $\mu$ DCG: 0.47	MRR: -0.0063 $\mu$ DCG: 0.0069
10-Q	2497	MRR: 0.12 $\mu$ DCG: 0.15	MRR: 0.16 $\mu$ DCG: 0.23	MRR: 0.11 $\mu$ DCG: 0.16	1416	MRR: 0.23 $\mu$ DCG: 0.31	MRR: 0.24 $\mu$ DCG: 0.34	MRR: 0.028 $\mu$ DCG: 0.05
SC 13G	1601	MRR: 0.2 $\mu$ DCG: 0.32	MRR: 0.25 $\mu$ DCG: 0.45	MRR: 0.13 $\mu$ DCG: 0.18	1395	MRR: 0.26 $\mu$ DCG: 0.42	MRR: 0.25 $\mu$ DCG: 0.42	MRR: -0.017 $\mu$ DCG: -0.0084
S-8	1482	MRR: 0.17 $\mu$ DCG: 0.21	MRR: 0.2 $\mu$ DCG: 0.26	MRR: 0.085 $\mu$ DCG: 0.12	1482	MRR: 0.22 $\mu$ DCG: 0.26	MRR: 0.22 $\mu$ DCG: 0.26	MRR: $-5.6 \times 10^{-5}$ $\mu$ DCG: 0.013

TABLE V: Results on FinanceBench: comparing *overall, not @k* metric values of gte-large (“Base”) to bi-enc(1) (“Exp”).

Type	Count	MRR Base	MRR Exp	MRR d	$\mu$ NDCG Base	$\mu$ NDCG Exp	$\mu$ NDCG d
10-K	112	$0.23 \pm 0.03$	$0.19 \pm 0.03$	-1.30	$0.52 \pm 0.02$	$0.52 \pm 0.02$	0.027
10-Q	15	$0.25 \pm 0.1$	$0.36 \pm 0.1$	1.00	$0.52 \pm 0.07$	$0.6 \pm 0.07$	1.100
8-K	9	$0.54 \pm 0.1$	$0.51 \pm 0.1$	-0.21	$0.8 \pm 0.1$	$0.83 \pm 0.1$	0.190
ECT	14	$0.38 \pm 0.1$	$0.39 \pm 0.1$	0.15	$0.78 \pm 0.09$	$0.81 \pm 0.08$	0.260
all	150	$0.27 \pm 0.03$	$0.25 \pm 0.03$	-0.63	$0.56 \pm 0.02$	$0.57 \pm 0.02$	0.530

## B. Qualitative Results

The first iteration yields the most gain. On in-domain filing types, the second iteration shows smaller effects and mixed results by filing type. On FinanceBench (OOD)  $\mu$ NDCG nudges up while MRR is mixed. To understand why, we analyzed “promoted passages”—query-chunk pairs where bi-enc(1) successfully retrieved relevant passages in the top position that bi-enc(0) missed in the top 5—

Examining these improvements across filing classes, we identified several main categories of promoted passages, including these three:

- **Table Retrieval** bi-enc(1) shows enhanced ability to retrieve business metrics from tables, likely reflecting the higher frequency of tables in SEC filings versus general text used as training data for bi-enc(0).
- **Technical Exact Match** bi-enc(1) learned to appropriately weight domain-specific keyphrases (e.g. “par value”) in financial contexts.
- **Semantic Understanding** bi-enc(1) recognized conceptual relationships without lexical overlap—connecting

“raising capital” (in  $q$ ) with “notes payable...lines of credit” (in  $c$ ) or “risks that could impact the company’s business” (in  $q$ ) with factors “adversely affecting capital and credit markets” (in  $c$ ).

For the three classes (10-Q, 8-K, ECT) in the FinanceBench dataset for which bi-enc(1) had improved aggregate metrics over bi-enc(0), we observed similar categories of promoted passages. For the final class, 10-K, we attempted to diagnose the lack of improvement by examining the inverse phenomenon of “demoted (relevant) passages”. We thereby observed several patterns prevalent in the (human) SME-written queries of FinanceBench which are likely not adequately represented in our training data, for example: compound queries asking for several things at once; queries mentioning accounting concepts such as “quick ratio” which need to be unpacked, using external or domain knowledge, into multiple queries for proper retrieval; queries containing “distractor” terms which degrade precision. We will propose methods of addressing this gap in Section VIII, below.

In the context of the in-domain SEC filings corpus we define

TABLE VI: List of Example Passages with Improvements in Experimental Model on Filings and Finance Bench Corpora

Pattern Name	Query	Promoted Passage	Class
technical exact match	What is the par value of the company's common stock?	In connection with the Reverse Stock Split, there was no change in the par value per share of \$0.001.	10-K
	What is the interest rate range that the company will pay on its indebtedness under the Revolving Credit Facility for the current period?	.. and any outstanding loans under the Revolving Facility will bear interest at either an Adjusted Term SOFR plus a margin of 1.00% to 1.75% or an Adjusted Base Rate plus a margin of 0% to 0.75%.	
	Which region had the worst topline performance for MGM during FY2022?	Las Vegas Strip Resorts Net revenues of \$8.4 billion in the current year compared to \$4.7 billion in the prior year, an increase of 77% ...	
semantic understanding	What steps is the company taking to mitigate the risk of not being able to raise the necessary capital to execute its business strategy?	Management plans to address the concerns, as needed, by (a) utilizing recent financing obtained through notes payable; (b) utilizing current lines of credit	10-K
	What potential risks and uncertainties does the company's CEO believe could impact the company's business due to current market conditions and geopolitical conflicts?	Furthermore, the capital and credit markets may be adversely affected by regional conflicts around the world and the possibility of a wider global conflict, global sanctions imposed in response to regional conflicts or an energy crisis.	10-Q
table retrieval	What is the change in the company's CEO's total compensation from the previous year?	For more information, please refer to the "Compensation Discussion and Analysis," as well as the "Narrative Disclosure to Summary Compensation Table and Grants of Plan-Based Awards Table."Name and Principal PositionFiscal YearSalary (\$)Bonus (\$)Stock Awards (\$)Non-equity Incentive Plan Compensation (\$)All Other Compensation (\$)Total (\$)Johanna 'Hanneke' Faber(4)Chief Executive Officer2024422,075 2,679,676 2,920,689 675,000 320,306 7,017,746...	DEF-14A
	How much was the Real change in Sales for AMCOR in FY 2023 vs FY 2022, if we exclude the impact of FX movement, passthrough costs and one-off items?	Net income attributable to Amcor 109 109 109 7.3 181 181 181 12.3 Net income attributable to non-controlling interests 3 3 4 4 Tax expense 103 103 68 68 Interest expense, net 35 35 70 70 Depreciation and amortization 145 144 EBITDA, EBIT, Net income and EPS 395 250 109 7.3 467 323 181 12.3 ...	ECT (Financebench)
	Does 3M have a reasonably healthy liquidity profile based on its quick ratio for Q2 of FY2023? If the quick ratio is not relevant to measure liquidity, please state that and explain why.	Inventories Finished goods 2,526 2,497 Work in process 1,527 1,606 Raw materials and supplies 1,227 1,269 Total inventories 5,280 5,372 Prepaids 674 435 Other current assets 539 456 Total current assets 15,754 14,688 Current liabilities Short-term borrowings and current portion of long-term debt \$ 3,033 \$ 1,938 Accounts payable 3,231 3,183 Accrued payroll...	10-Q (Financebench)

a “promoted passage” (with respect to query  $q$ ) as a  $c \in C(d)$  such that

- 1)  $r_{q,c} = 4$
- 2)  $c \in C_d(q, 0, 1) - C_d(q, 5, 0)$ ,

that is  $c$  is (within  $d$ ) the closest passage to  $q$  according to bi-enc(1), but would not be retrieved within the top 5 under bi-enc(0). One major pattern seen when the queries asks for relatively straightforward business metrics, the promoted passage is often a (section of a) table, indicating that the fine-tuned model does a better job of recognizing when to retrieve tables than the baseline model. Another common pattern that is seen is that when a technical financial reporting term (example “*par*” as in “*par* value”) has an exact match in the “promoted

passage”. This leads us to speculate that the fine-tuning causes the model to act more like an exact-keyword match in certain respects, that is to weigh more highly exact matches of such technical terms. Conversely, the fine-tuned model on occasion appears to exhibit greater semantic understanding of the query than the out-of-box model, as in the case of this example: “What steps is the company taking to mitigate the risk of not being able to raise the necessary capital to execute its business strategy?”. One promoted passage for this query is “...Management plans to address the concerns, as needed, by (a) utilizing recent financing obtained through notes payable; (b) utilizing current lines of credit.” Although the concerns are not explicitly identified within the passage as being about being able to raise capital, an expert reader, and also the model

TABLE VII: List of Example Passages Showing Negative Experiment Results on FinanceBench 10-K Subcorpus

Pattern Name	Query	Demoted Passage
Multi-faceted Question	Among operations, investing, and financing activities, which brought in the most (or lost the least) cash flow for AMD in FY22?	assets (1,197) (920) (231) Payables to related parties 379 7 (135) Accounts payable 931 801 (513) Accrued liabilities and other 546 526 574 Net cash provided by operating activities 3,565 3,521 1,071 Cash flows from investing activities: Purchases of property and equipment (450) (301) (294)
Terminologically Dense Query	Does AMD have a reasonably healthy liquidity profile based on its quick ratio for FY22? If the quick ratio is not relevant to measure liquidity, please state that and explain why?	Current liabilities Accounts payable \$ 2493 \$ 1,321 Accumulated deficit (131) (1451) Accumulated other comprehensive loss (41) (3) Total stockholders equity 54,750 7,497 Common stock, par value \$ 0.01 shares authorized 2250, shares issued 1645 and 1232
Time-Specific Information Request	Has CVS Health reported any materially important ongoing legal battles from 2022, 2021 and 2020	The Company is named as a defendant in a number of lawsuits The Company is facing multiple lawsuits, including by state Attorneys General, governmental subdivisions and several putative class actions, regarding drug pricing and its rebate arrangements with drug manufacturers.
Label noise	Assume that you are a public equities analyst. Answer the following question by primarily using information that is shown in the balance sheet: what is the year end FY2018 net PPNE for 3M? Answer in USD billions.	Notes to Consolidated Financial Statements are an integral part of this statement.

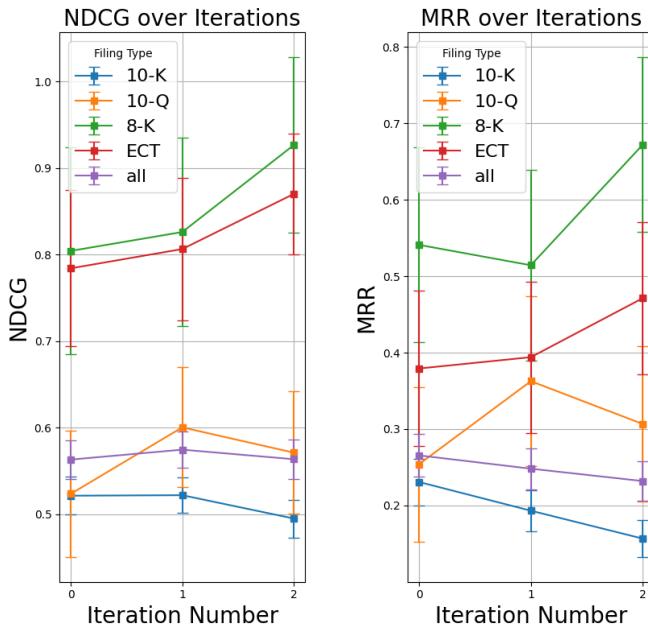


Fig. 3: Metrics with standard errors as measured on FinanceBench Evaluation Set

are able to infer from the nature of the plans to address the concerns that these concerns are in fact related to raising capital. More examples can be found in Table VI for the filings corpus and FinanceBench, respectively.

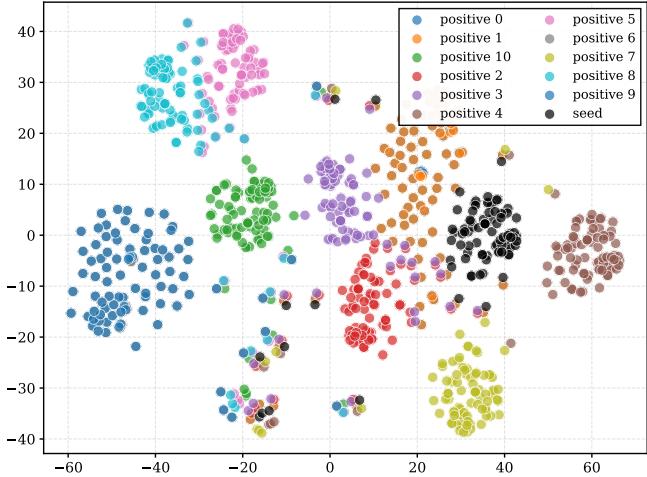
Before explaining “promoted passage” and “demoted passage” for FinanceBench we explain how we propagated labels from the FinanceBench dataset to our chunks. FinanceBench is mainly a Question Answering dataset rather than a relevance dataset, but it has certain passages (not necessarily lining up in any straightforward way with our chunks, and of very different lengths) marked as “evidence” chunks for the answer. Up to

a first approximation, we consider a chunk in our chunking over the FinanceBench corpus (the documents which occur as documents in the evaluation set) relevant if and only if the chunk intersects at least one evidence chunk (considered as a character span in the original document). There is one major adjustment to this simple picture: if an intersection is very “minor”, then the chunk is not considered relevant. To define what we mean by “minor”, denote the chunk in question by  $c$  and the evidence passage by  $p$ . Then (by definition) the intersection is “minor” if  $\text{len}(c \cap p) \leq \frac{1}{3} \min(\text{len}(c), \text{len}(p))$ . Note that this can occur only if  $c$  is the first or last chunk to intersect  $p$ , because an “interior” chunk of the intersection will intersect along its entire length. This was done to decrease the label noise, or false positives, in which a chunk had a very short and insignificant overlap with the evidence passage, but was truly irrelevant to the query. In spite of this provision, there was still a certain amount of label noise, according to the manual inspection of the results, as shown in the table below, and this may have an impact on the reported aggregate metrics considering the modest size of the dataset.

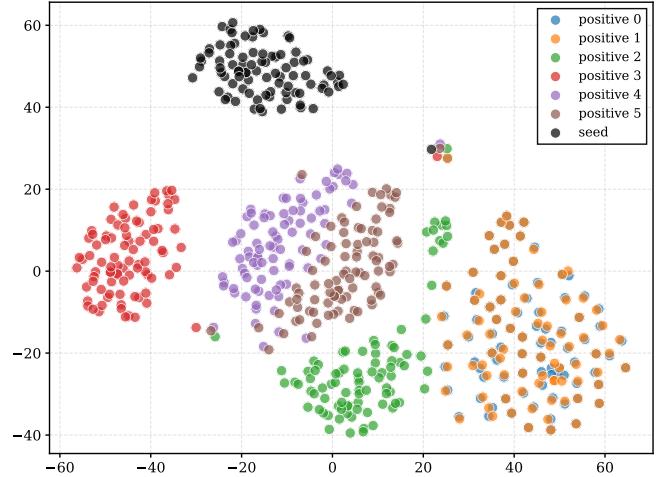
Because every chunk in FinanceBench is labeled with a relevance, unlike in the case of our filings dataset, where only a sample (including the top  $k$  ranks under the models) is labeled, we are able to calculate *overall* NDCG and MRR, not just versions of these metrics  $@k$ . For the same reason, in the context of the Financebench dataset, we can use a more natural and symmetrical definition of *promoted* and *demoted* passages. Namely, let  $q$  and  $c \in C(d)$  be fixed with  $r_{q,c} = 4$ , so that  $c$  is relevant to  $q$ . Denote by  $\rho_j(c)$  the rank of  $c \in C(d)$  under  $\text{bi-enc}(j)$ . Consider

$$\Delta_i(c) := \rho_i(c) - \rho_{i+1}(c).$$

Noting that ranks are numbered in an increasing fashion from closest to furthest from  $q$ , we define a promoted (resp. demoted) passage  $c$  as one for which  $\Delta_i(c) > \mu$  (resp.  $< \mu$ ),  $\mu > 0$  some margin. After sorting all of  $C(D_{\text{test}})$  by the values of  $\Delta_i(c)$  in



(a) Q1: "What is the regulatory body that regulates the company?"



(b) Q2: "What is the company's current research focus in terms of product development?"

Fig. 4: Plots of  $t\text{-SNE}(\mathcal{S}(q))$ , the low-dimensional projections of difference vectors of positive and negative chunks, with  $t\text{-SNE}(\mathcal{S}_{\text{seed}}(q))$  in black, highlighting the benefits of positive example mining in obtaining a more diverse training set.

descending order, we define the “most promoted” (resp. “most demoted”) passages for a corpus as the head  $C(D_{\text{test}})[:-50]$  (resp. tail  $C(D_{\text{test}})[-50 :]$ ), and a selection of these is what we actually present in Tables VI and VII.

## VII. ANALYSIS OF POSITIVE EXAMPLE MINING BENEFITS

Intuitively, the bi-encoder model learns by contrasting embeddings of passages judged relevant versus irrelevant for the same query, effectively focusing on difference vectors between these pairs. The more diverse the empirical distribution of such differences we expose during training, the more robustly it can internalize the semantics of the query. When relying only on an inPars-style sampling strategy (as in Anderson et al. [3]), the training signal is restricted to a narrow subset anchored on a single seed positive, which we visualize as black points. Augmenting training with mined positives from our method introduces many additional, complementary difference directions (colored points), broadening coverage across the space. In low-dimensional visualizations, the inPars-only subset typically occupies approximately a single cluster, whereas the augmented set spans multiple clusters; collectively, this expanded, multi-modal support helps explain the superior performance of our approach.

Each plotted training point is a difference vector:

$$\mathbf{d}(q; c_{\text{rel}}, c_{\text{irrel}}) = \mathbf{c}_{\text{rel}} - \mathbf{c}_{\text{irrel}}.$$

In Figure 4 we visualize all differences where the positive is highly relevant and the negative is less relevant:

$$\mathcal{S}(q) = \{\mathbf{c}_{\text{rel}} - \mathbf{c}_{\text{irrel}} \mid r(q, c_{\text{rel}}) = 4, r(q, c_{\text{irrel}}) < 3\}.$$

The inPars-only subset anchored on the seed positive (plotted in black):

$$\mathcal{S}_{\text{seed}}(q) = \{\mathbf{c}_{\text{rel}} - \mathbf{c}_{\text{irrel}} \in \mathcal{S}(q) \mid c_{\text{rel}} = c_{\text{seed}}\}.$$

For visualization we use a two-dimensional t-SNE embedding; black points denote the inPars-only subset and colored points are additional mined positives:

$$t\text{-SNE}(\mathcal{S}(q)).$$

A commonly used metric in literature, t-SNE [12] is a dimensionality reduction technique that is particularly well-suited for visualizing high-dimensional data, as it preserves local structures when projecting to lower-dimensional spaces. For the plots, we select two exemplary queries: (a) "What is the regulatory body that regulates the company?" (Figure 4a) and (b) "What is the company's current research focus in terms of product development?" (Figure 4b), both associated to the 10-K document class.

## VIII. CONCLUSIONS, FUTURE WORK

We showed that an iterative pipeline using mining of positive and negative examples from a large corpus helps gather training data for domain adaptation of a retrieval model, though most of the benefits to aggregate metrics come from a single iteration. The main limitations of the study are first that passage identification via LLM relies on the current best model with sampling heuristics, missing many important aspects. Second, that the only public benchmark (as of time of writing) in this area, namely FinanceBench, while of high quality, is relatively small. This demonstrates that query expansion methods are needed either using knowledge graph methods, e.g., “GraphRAG” [8], or integrating external knowledge bases. Alternatively, “agentic” models with reasoning capabilities could create more specific queries better suited as anchors in bi-encoder based retrieval. The agentic model could also examine retrieval results to plan better candidate retrieval from the corpus through tool use, query augmentation, and improved heuristics. Rather than capping the investigation into domain

adaptation of retrieval models, we believe this work establishes a solid foundation for broader exploration in the indicated directions.

## REFERENCES

- [1] Meta AI. Llama-3.1-70b-instruct model card. <https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct>, 2024. Retrieved January 15, 2025.
- [2] Artificial Analysis. Artificial analysis ai review 2024 highlights. <https://artificialanalysis.ai/downloads/ai-review/2024/Artificial-Analysis-AI-Review-2024-Highlights.pdf>, 2024. Retrieved January 15, 2025.
- [3] Peter Anderson, Mano Vikash Janardhanan, Jason He, Wei Cheng, and Charlie Flanagan. Greenback bears and fiscal hawks: Finance is a jungle and text embeddings must adapt. In Franck Dernoncourt, Daniel Preoțiuc-Pietro, and Anastasia Shimorina, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 362–370, Miami, Florida, US, November 2024. Association for Computational Linguistics.
- [4] Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2387–2392, 2022.
- [5] CapTide. How to do agentic rag on sec edgar filings, 2024. URL <https://www.captide.co/insights/how-to-do-agentic-rag-on-sec-edgar-filings>. Accessed March 5, 2025.
- [6] Jaeyoung Choe, Jihoon Kim, and Woohwan Jung. Hierarchical retrieval with evidence curation for open-domain financial question answering on standardized documents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16663–16681, 2025.
- [7] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, Hillsdale, NJ, 2nd edition, 1988. ISBN 0-8058-0283-5.
- [8] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.
- [9] Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. Financebench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*, 2023.
- [10] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.
- [11] Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 881–893, 2024.
- [12] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [13] Bo Peng, Emmanuele Chersoni, Yu-Yin Hsu, Chu-Ren Huang, et al. Is domain adaptation worth your investment? comparing bert and finbert on financial tasks. In *Proceedings of the Third Workshop on Economics and Natural Language Processing*, pages 37–44, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics (ACL).
- [14] Varshini Reddy, Rik Koncel-Kedziorski, Viet Dac Lai, Michael Krumdick, Charles Lovering, and Chris Tanner. DocFinQA: A long-context financial reasoning dataset. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 445–458, Bangkok, Thailand, August 2024. Association for Computational Linguistics.
- [15] Yixuan Tang and Yi Yang. Do we need domain-specific embedding models? an empirical investigation. *arXiv preprint arXiv:2409.18511*, 2024.
- [16] Yixuan Tang and Yi Yang. Finmteb: Finance massive text embedding benchmark. *arXiv preprint arXiv:2502.10990*, 2025.
- [17] U.S. Securities and Exchange Commission. *EDGAR Filer Manual Volume II: EDGAR Filing*. U.S. Securities and Exchange Commission, sep 2025. URL <https://www.sec.gov/edgar/filer-manual>.
- [18] U.S. Securities and Exchange Commission. Edgar - electronic data gathering, analysis, and retrieval system, 2025. URL <https://www.sec.gov/edgar.shtml>.
- [19] Shuting Wang, Jiejun Tan, Zhicheng Dou, and Ji-Rong Wen. Omnieval: An omnidirectional and automatic rag evaluation benchmark in financial domain. *arXiv preprint arXiv:2412.13018*, 2024.
- [20] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020.
- [21] Yiyun Zhao, Prateek Singh, Hanoz Bhathena, Bernardo Ramos, Aviral Joshi, Swaroop Gadiyaram, and Saket Sharma. Optimizing LLM based retrieval augmented generation pipelines in the financial domain. In Yi Yang, Aida Davani, Avi Sil, and Anoop Kumar, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 279–294, Mexico City, Mexico, June 2024. ACL.