

# Greenback Bears and Fiscal Hawks: Finance is a Jungle and Text Embeddings Must Adapt

Peter Anderson Mano Vikash Janardhanan Jason He Wei Cheng Charlie Flanagan  
Balyasny Asset Management

{peanderson, mjanardhanan, jhe, wcheng, cflanagan}@bamfunds.com

## Abstract

Financial documents are filled with specialized terminology, arcane jargon, and curious acronyms that pose challenges for general-purpose text embeddings. Yet, few text embeddings specialized for finance have been reported in the literature, perhaps in part due to a lack of public datasets and benchmarks. We present BAM embeddings, a set of text embeddings finetuned on a carefully constructed dataset of 14.3M query-passage pairs including both public and proprietary financial documents. Demonstrating the benefits of domain-specific training, BAM embeddings achieve Recall@1 of 62.8% on a held-out test set, vs. only 39.2% for the best general-purpose text embedding from OpenAI. Further, BAM embeddings increase question answering accuracy by 8% on FinanceBench and show increased sensitivity to the finance-specific elements that are found in detailed, forward-looking and company and date-specific queries. To support further research we describe our approach in detail, quantify the importance of hard negative mining and dataset scale, and publicly release our embeddings<sup>1</sup>.

## 1 Introduction

Portfolio managers and analysts have access to millions of financial documents. Text embeddings are a key component of the information retrieval and retrieval-augmented generation (RAG) systems (Lewis et al., 2020) that can help extract insights from this mass of information. However, the financial domain poses unique challenges for text embeddings. Financial documents are filled with specialized terminology (‘par value’, ‘stagflation’), jargon (‘Chinese wall’), curious acronyms (‘CAGR’, ‘DCF’, ‘VIX’), and technical terms and company names that collide with ordinary words (‘short’, ‘forward’, ‘spread’, ‘Apple’, ‘Stripe’). Is a CDO

similar to a CFO? And what is a Greenback bear<sup>2</sup>?

Despite their importance, few text embeddings specialized for finance have been reported in the literature. To address this gap, we present and publicly release BAM embeddings, a set of text embeddings optimized for financial document retrieval. BAM embeddings are based on Multilingual-E5 (Wang et al., 2024b), further finetuned on a carefully filtered, clean dataset of 14.3M query-passage pairs (6B tokens) constructed from 2.8M public and proprietary financial documents. While we cannot release our dataset, we describe in detail our data curation and query generation strategy, finetuning process, and approach to deployment in a high-priority application.

On a held-out set of 447K query-passage pairs, BAM embeddings achieve Recall@1 of 62.8%, far surpassing the Multilingual-E5 base model (34.3%) as well as large closed-source models (e.g., OpenAI’s 3072-dim text-embedding-3-large model, 39.2%). Quantitatively, we show that hard negative mining (+5.3%) and data scale (+4.5%) are critical to achieving this performance. Deploying BAM embeddings in an application alongside traditional lexical search (Okapi BM25), we find that BAM embeddings outperform lexical search over all query lengths. Notably, vector search with BAM embeddings improves as queries become longer and more detailed, while lexical search degrades.

Finally, we evaluate BAM embeddings in a public RAG benchmark using FinanceBench (Islam et al., 2023). Replacing OpenAI’s ada-002 embeddings with ours increases question answering correctness by 8%. Qualitatively, we observe that after finetuning, embeddings are more sensitive to company names, tickers and financial metrics, leading to improved performance detailed, forward-looking, and company or date-specific queries.

<sup>2</sup>A Greenback bear is an investor who believes the US dollar will decline in value. A fiscal hawk argues for a reduction in government spending.

<sup>1</sup><https://huggingface.co/BalyasnyAI/multilingual-e5-base>

## 2 Related Work

**Financial LLMs** Much of the previous work in Financial NLP has focused on adapting large language models (LLMs) to finance. FinBERT (Yang et al., 2020) was the first LLM to demonstrate the benefits of pretraining on company reports, broker research and earnings transcripts. When finetuned for financial sentiment classification, FinBERT outperformed the standard BERT model pretrained on generic text (Devlin et al., 2019). FLANG (Shah et al., 2022) demonstrated that the gains from pretraining on financial documents extended to other downstream tasks, including news headline classification, named entity recognition (NER), structure boundary detection, and question answering in the finance domain. Compared to these 100M parameter models, the 50B parameter BloombergGPT model (Wu et al., 2023) facilitated evaluation via few-shot prompting rather than finetuning, again outperforming general-purpose LLMs. Continuing the trend towards chat-based approaches, PIXIU (Xie et al., 2023) developed a financial instruction dataset and benchmark generalizing 9 financial NLP tasks, and used it to finetune LLaMA (Touvron et al., 2023).

**Domain-Specific Text Embeddings** Even with pretraining or finetuning for finance, LLMs require up-to-date information, which is typically retrieved using text embeddings (Lewis et al., 2020). Domain-specific embeddings have been well-studied in healthcare (Alsentzer et al., 2019; Lee et al., 2019) and law (Chalkidis et al., 2020). Surprisingly, no text embeddings specialized for financial document retrieval have been reported in the literature, perhaps in part due to a lack of public datasets and benchmarks on which to train and evaluate. Standard benchmarks such as MTEB (Muennighoff et al., 2022) contain no company reports, broker research or earnings transcripts<sup>3</sup>, while finance-oriented benchmarks such as FinanceBench (Islam et al., 2023) and PIXIU (Xie et al., 2023) are designed to evaluate LLMs not text embeddings. We address the lack of finance text embeddings by releasing BAM embeddings. Similar to Ma et al. (2021); Cho et al. (2022) and others, our approach relies on synthetic query generation for training data.

<sup>3</sup>The most relevant MTEB datasets – FIQA 2018 (Maia et al., 2018) and Financial PhraseBank (Malo et al., 2014) – are based on financial news and blog posts.

## 3 Dataset

To finetune and evaluate BAM embeddings, we construct a dataset of 15.2M *query-passage* pairs. Text *passages* are drawn from public and proprietary documents (refer Section 3.1). For each passage, a matching *query* is generated using a few-shot prompted LLM (refer Section 3.2).

### 3.1 Sampling Text Passages

**Raw Documents** Text passages are sourced from 2.8M documents published in the two-year period ending 31 March 2024, comprising:

- Equity Research: 1.7M broker reports
- Macro Research: 430K broker reports
- Sales Commentary: 94K emails from broker sales desks to buy-side clients
- Company Transcripts: 102K public transcripts from earnings calls and investor days
- EDGAR Filings: 32K public Form 10-K and Form 10-Q company filings
- Interview Transcripts: 102K interviews between investors and industry experts
- Notes and Previews: 377K proprietary notes and earnings previews written by buy-side analysts and portfolio managers

**Parsing and Splitting** Most documents are stored in PDF format. We convert them to text using an internal PDF-to-text conversion tool, then split the text into passages. Our splitting strategy recursively splits text based on a generic list of typical paragraph separators (‘\n\n’, ‘\n’, etc), while attempting to avoid splitting questions and related answers (which are common in Company Transcripts and Interview Transcripts, and clearly labelled). We choose a maximum passage length of 512 tokens (350–400 words). This typically provides sufficient context to understand the text while shorter passages can be difficult to interpret. We use regular expressions and heuristics to remove legal disclosures and most tables from the dataset (we focus on text passage retrieval and leave information extraction from tables to future work).

**Document Context** After splitting, text passages are frequently missing crucial information such as the name and stock ticker of the company referenced in the text, and the date. We augment each text passage by prepending one line of document context. The content of this context line differs by document type — for Company Transcripts it contains the company name, ticker, and the event

cotton corn production cost trends
wet wipes new products
sector rotation this week
Reliance CapEx
artificial intelligence at medtronics
How has brexit affected supply chain operations and companies?
Chinese real estate new residential sales
beazley price
What is going on with pulpwood costs?
How is Teck Resources generating shareholder value through structured separation?

Table 1: Examples of the human-written queries used to seed synthetic queries via few-shot prompting.

(e.g., ‘FY23 earnings call’); for Interview Transcripts it contains the interviewee’s name and job title and the title of the interview. We use the same approach when finetuning the embeddings and in deployment.

### 3.2 Query Generation

Rather than finetuning on user-generated queries, we choose to synthetically generate all queries in the dataset. Synthetic generation is highly scalable, eliminates privacy concerns, ensures complete coverage of all text passages in the corpus, and enables finetuning and evaluating on queries that are longer and more complex than user queries.

**Few-Shot Examples** To seed our query generation strategy, we randomly sample text passages and ask a group of quantitative researchers, engineers and product managers to write a query for each passage. Annotators are instructed that the passage should be a top result for that query in a world-class document retrieval system (refer Appendix). Using this approach, we collect 231 passage-query pairs to use for few-shot prompting (see Table 1 for randomly selected examples).

**LLM Query Generation** We prompt an LLM to generate a single matching query for every text passage in the corpus, or output ‘SKIP’ if a query can’t be generated. To encourage quality and diversity, two passages with human-written queries are randomly selected and included as few-shot examples (refer to the Appendix for the complete prompt). Trading off computational cost and query quality, we use the Mistral 7B Instruct model (Jiang et al., 2023) for query generation. Even using vLLM (Kwon et al., 2023) for high throughput, several weeks of A100 gpu time are required to generate queries for the entire dataset. In initial experiments we also used a second LLM call to paraphrase the generated queries for increased diversity and chal-

Document Type	Query-Passage Pairs
Equity Research	4,512K
Macro Research	954K
Sales Commentary	596K
Company Transcripts	2,603K
EDGAR Filings	2,640K
Interview Transcripts	2,408K
Notes & Previews	1,447K
Total	15,159K

Table 2: Dataset composition by document type.

lenge, but found this refinement was less effective than simply generating more queries.

**Filtering** Passages for which the LLM failed to generate a valid query are identified by the presence of the special ‘SKIP’ token provided in the prompt, or by responses that begin with identified phrases that indicate failure such as ‘no query’, ‘no question’ or ‘understood’. These are removed from the dataset, along with all duplicate queries, which arise when multiple text passages generate the same query, e.g., ‘Apple 2024 EPS’. As illustrated in Table 2, after filtering (including filtering performed during hard negative mining, refer Section 4) the final dataset consists of 15.2M query-passage pairs, which are separated into train-val-test splits containing 14.3M, 444K, and 447K examples, respectively. All passages from the same document are assigned to the same split. This avoids contaminating the val or test splits if documents are repetitive.

**Query Realism** We do not aim to replicate the query distribution from our legacy document retrieval system, which are typically short, simple keyword queries. These are shaped by user interactions with a weaker BM25-based system. Instead, we aim to support longer, more complex queries, which we now encourage (refer Section 5.3).

## 4 BAM Embeddings

**Baseline Model** We finetune the Multilingual-E5 model (Wang et al., 2024b), which is pretrained on 1B multilingual text pairs and pre-finetuned on a combination of general-purpose labeled datasets. Multilingual-E5 embeddings are based on the XLM-RoBERTa architecture (Conneau et al., 2019) which has three sizes: *small* (118M model params, 384-dim embeddings), *base* (278M model params, 768-dim embeddings), and *large* (560M model params, 1024-dim embeddings). We finetune small and base models; in preliminary experiments we

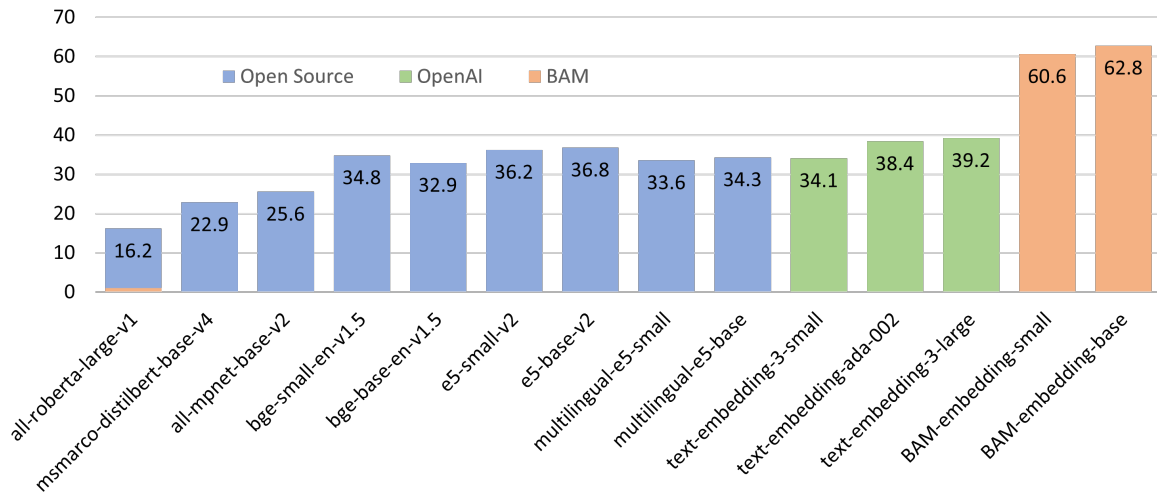


Figure 1: Passage retrieval results: Recall@1 on a held-out test split of 447K query-passage pairs. BAM embeddings finetuned for financial document retrieval significantly outperform general-purpose embeddings.

found that the large model performed no better than the base model so we discontinued finetuning.

We select Multilingual-E5 because the model achieved competitive baseline retrieval performance on our dataset, and because the embedding is mean-pooled<sup>4</sup>. This enables us to calculate a contextualized embedding for any sentence within a passage by simply mean-pooling over the relevant subset of tokens. In our application, we exploit this affordance to highlight the sentences in a document that are most relevant to a user’s query, based on the similarity between each sentence and the query. Embedding models based on the output of the CLS token, such as BGE (Xiao et al., 2023), do not offer this capability.

**Finetuning** We use the standard InfoNCE contrastive loss (van den Oord et al., 2019) that requires the model to identify the positive passage for each query from a set of negative passages. Negative passages are comprised of in-batch negatives (the other passages in the same minibatch) plus 3 hard negatives per query (see below).

We finetune for 3 epochs using a batchsize of 512 and initial learning rate  $\{3, 2, 1\} \times 10^{-5}$  for the {small, base, large} models. Following Wang et al. (2024a), during finetuning, evaluation and in deployment, we add the prefixes ‘query: ’ and ‘passage: ’ to queries and passages respectively, allowing the model to better represent short queries and long passages in the same embedding space.

<sup>4</sup>Mean-pooling is applied to the output of the final transformer layer across all tokens in the query or passage.

**Hard Negative Mining** Hard negative mining improves the quality of learned embeddings by introducing negative examples that are more challenging to detect than randomly-sampled negatives (Gao et al., 2021; Karpukhin et al., 2020). For each query, we identify 3 hard negative passages using an early version of our model finetuned with 37% of the full dataset and no hard negatives. Specifically, we embed all 15.2M queries and passages in the dataset, retrieve the top 1K passages for each query, and label the passages ranked 200–202 places lower than the positive passage as hard negatives. This hyperparameter was set after testing several different options in initial experiments. If the positive passage is not in the retrieved passages, the query-passage pair is removed from the dataset, as manual inspection indicates that these are typically low-quality pairs.

## 5 Results and Analysis

### 5.1 Retrieval

We evaluate the retrieval performance of BAM embeddings using the held-out test set of 447K query-passage pairs. We benchmark against 12 other models, including the E5 (Wang et al., 2024a) and BGE (Xiao et al., 2023) families of models, and text embeddings from OpenAI. Since each query has only a single correct passage, we report Recall@ $K$  with  $K = 1, 10, 50$  rather than NDCG. Since the users of our system (both human and LLM agent) have limited attention, we focus on Recall@1 although the same trends hold across other values of  $K$ .

As illustrated in Figure 1, finetuning the 768-dim

Embedding	Dim	Passage Retrieval			Document Retrieval		
		R@1	R@10	R@50	R@1	R@10	R@50
text-embedding-3-small	1536	34.1	68.3	84.4	61.1	85.0	90.1
text-embedding-ada-002	1536	38.4	71.6	85.7	64.3	86.2	91.3
text-embedding-3-large	3072	39.2	73.7	87.8	64.9	87.4	91.8
multilingual-e5-small	384	33.6	67.1	82.1	61.2	84.2	89.7
multilingual-e5-base	768	34.3	68.2	83.1	62.6	85.6	91.1
BAM-embedding-small	384	60.6	89.6	96.6	81.4	95.5	97.9
BAM-embedding-base	768	<b>62.8</b>	<b>91.0</b>	<b>97.3</b>	<b>83.0</b>	<b>96.3</b>	<b>98.4</b>
...1 hard negative, full training data	768	61.8	90.4	<b>97.3</b>	82.2	96.1	<b>98.4</b>
...No hard negatives, full training data	768	57.5	88.8	97.2	79.5	95.8	98.5
...No hard negatives, 37% training data	768	53.0	86.1	96.1	76.4	94.6	97.9

Table 3: Base-sized BAM embeddings outperform the much larger OpenAI embeddings (top panel) and the baseline model (middle panel) on all recall metrics. Ablation studies (bottom 3 rows) highlight the importance of hard negative mining and scaling the finetuning data.

Multilingual-E5 base model improves Recall@1 from 34.3% to 62.8% – surpassing an array of open-source and close-source models including OpenAI’s 3072-dim text-embedding-3-large model, which achieves 39.2% Recall@1 (with a much larger embedding). We observe similar gains with the small model, which achieves 60.6% Recall@1.

**Ablation Studies** In Table 3 we report Recall@K for both passage retrieval and document retrieval (i.e., retrieving any passage from the document containing the correct passage). We compare to OpenAI embeddings (top panel) and the baseline models (middle panel). We find that base-sized BAM embeddings perform best on every metric, and hard negative mining and data scale are crucial to achieving this performance.

Without hard negative mining, passage Recall@1 for the base-sized model drops from 62.8% to 57.5%, although 1 hard negative is sufficient to capture most of the benefits (61.8% vs. 62.8% with 3 hard negatives). Reducing the amount of training data to 37% of the full dataset (representing only one year of data, and 4 document types instead of 7) further reduces Recall@1 from 57.5% to 53.0%, demonstrating the value of scaling the dataset over a large document corpus.

**Qualitative Analysis** In Table 4 we provide an example of how query similarity changes before and after finetuning. For more general insights, we randomly select 100 queries with a large improvement in recall under the finetuned model, and 100 queries with no improvement, and ask ChatGPT to identify qualitative differences between the queries.

According to ChatGPT, the queries that improved the most are company-specific (focused

on individual companies and particular quarters or fiscal years), forward-looking (referencing future projects and growth), searching for specific financial metrics (such as PE ratios or adjusted net income), and phrased as questions rather than statements. Queries about general financial and economic concepts, such as ‘capital issuance’, improved the least.

## 5.2 Results on FinanceBench

Given the lack of public benchmarks for financial document retrieval, in this section we report results on FinanceBench (Islam et al., 2023), a benchmark for financial question answering. We co-opt FinanceBench’s retrieval-augmented generation (RAG) setting to assess how retrieval with different text embeddings affects answer accuracy.

We focus on the Shared Vector Store setting, in which text passages from a collection of 368 10-K and 10-Q reports are stored in a single vector database, which is queried by an LLM to answer 150 questions derived from those documents. Islam et al. (2023) report that GPT-4 correctly answers only 19% of questions using OpenAI ada-002 text embeddings. However, the original RAG pipeline has several weaknesses. We improve it by:

1. Using an LLM to rewrite the question before querying the vector store. This eliminates distracting text containing formatting instructions
2. Prepending the filename of the parent document to the beginning of each text passage, which preserves document context including the company name and the filing date
3. Prompting the LLM to generate concise answers (which are more in keeping with the

Query: ASMI's price-to-earnings ratio

**Nearest Neighbors Before Finetuning:**

1. ASMI - earnings report analysis
2. ASMedia's share price
3. ASMI revenue goals
4. Aleoa's price to earnings ratio
5. Aviva's cost-to-income ratio
6. Axis Bank's cost-to-income ratio

**Nearest Neighbors After Finetuning:**

1. ASMI - earnings report analysis
2. ASMI outlook
3. What is the current market outlook for ASMI?
4. What is the rating and PE ratio of ASMI?
5. What is ASMI's revenue growth forecast for 2024?
6. Interesting read on ASMI stock.

Table 4: Nearest neighbor queries before and after finetuning Multilingual-E5. Before finetuning, embeddings capture too much lexical similarity, e.g. ASMI is similar to ASMedia; price-to-earnings is similar to cost-to-income. After finetuning, embeddings are more sensitive to tickers and stock names.

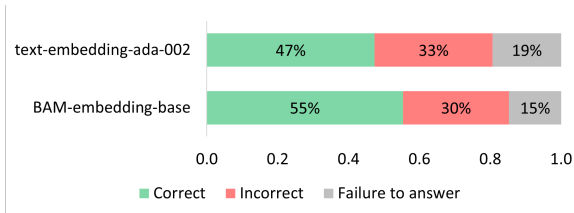


Figure 2: FinanceBench results under the Shared Vector Store setting. Replacing OpenAI ada-002 embeddings with BAM embeddings increases accuracy by 8%.

gold answers and easier to evaluate)

4. Replacing GPT-4 with GPT-4o

Based on human evaluation, our improved RAG pipeline achieves 47% accuracy vs. 19% reported by Islam et al. (2023). As illustrated in Figure 2, replacing the 1536-dim ada-002 embeddings with 768-dim BAM embeddings improves accuracy further to 55% – even though most FinanceBench questions are table-based, and BAM embeddings were optimized for text passage retrieval. The remaining errors are mostly attributable to the LLM (extracting numbers from tables and calculating derived metrics such as operating margin).

### 5.3 Real-world Deployment

**Application** We have deployed BAM embeddings in a RAG service that indexes 5.7M financial documents (1.3TB of raw data), providing a backend API for 3 different frontend applications (two market intelligence and search platforms and a chatbot). We use OpenSearch because it supports approximate nearest neighbor vector search in conjunction with traditional filtering operations. Filters are used to restrict search results based on document date ranges, company tickers, and tags such as document type, event name, data vendor, etc. In addition to OpenSearch, we maintain a NoSQL database to store document fields and metadata that would add excessive overhead to OpenSearch.

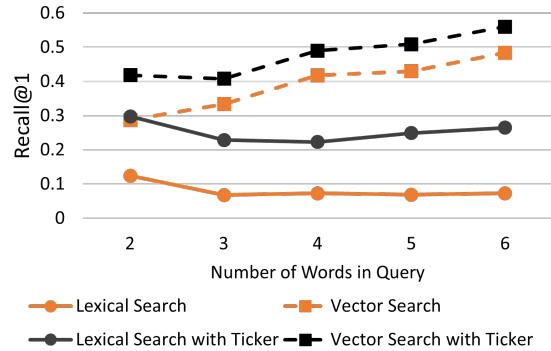


Figure 3: Comparison of vector search using BAM embeddings with lexical search (BM25). Vector search is superior to lexical search, and improves on longer and more detailed queries (while lexical search degrades).

**Weight Averaging** Before deploying BAM embeddings, we average the parameters of 5 finetuned checkpoints (trained for between 2.5 and 3 epochs) with the baseline model, with a 50% weighting on the baseline model and 10% weighting on each checkpoint. We are motivated by (Wortsman et al., 2022b,a) who show that averaging the weights of zero-shot and finetuned models improves accuracy and robustness to out of distribution queries. Robustness is an important consideration because we expect the distribution of user queries to drift over time (and may not perfectly match our generated queries to start with). Weight averaging sacrifices 1.1% Recall@1 on our dataset, but improves NDCG@10 on a representative out-of-domain dataset (FiQA 2018) by 2.2% compared to the final checkpoint.

**Comparison to Lexical Search** Our production document retrieval service provides an ideal opportunity to benchmark the performance of vector search (using BAM embeddings) against traditional lexical search (using OpenSearch's Okapi BM25 implementation). To quantify the impact of

query length, we benchmark queries containing 2–6 words, in each case using 1K randomly-selected company transcript queries from our test split. Date ranges are restricted to a one year period that contains the positive passage. We evaluate both with and without filtering on the correct stock ticker.

As illustrated in Figure 3, vector search outperforms lexical search in both settings, regardless of query length. While two-word queries are often ambiguous, vector search recall improves as queries become longer and more specific. In contrast, lexical search degrades on longer queries. This is consistent with our observation that users conditioned to using lexical search tools often limit their queries to 2 or 3 words. To encourage users to write longer queries, in our frontend application we implement query completion/autocomplete using high-quality examples from our dataset.

## 6 Conclusion

We release text embeddings specialized for finance, trained on a dataset of 14.3M synthetic queries constructed from public and proprietary financial documents. On a held-out test set, BAM embeddings achieve 62.8% Recall@1 vs. 39.2% for the best general-purpose text embeddings. On the FinanceBench benchmark, replacing general-purpose embeddings with ours improves question answering accuracy by 8%, demonstrating our dataset and model’s ability to generalize to out-of-domain settings. Finally, we show that in a production document retrieval service, BAM embeddings outperform BM25 over all query lengths, and (unlike BM25) retrieval improves on longer and more detailed queries.

## References

- Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. [Publicly available clinical BERT embeddings](#). In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. [LEGAL-BERT: The muppets straight out of law school](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.
- Sukmin Cho, Soyeong Jeong, Wonsuk Yang, and Jong Park. 2022. [Query generation with external knowledge for dense retrieval](#). In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 22–32, Dublin, Ireland and Online. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Unsupervised cross-lingual representation learning at scale](#). *Preprint*, arXiv:1911.02116.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#). *Preprint*, arXiv:1810.04805.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [SimCSE: Simple contrastive learning of sentence embeddings](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. [FinanceBench: A new benchmark for financial question answering](#). *Preprint*, arXiv:2311.11944.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Léo Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. [Dense passage retrieval for open-domain question answering](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

- Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. [BioBERT: a pre-trained biomedical language representation model for biomedical text mining](#). *Bioinformatics*, 36(4):1234–1240.
- 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. *NeurIPS*.
- Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. 2021. [Zero-shot neural passage retrieval via domain-targeted synthetic question generation](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1075–1088, Online. Association for Computational Linguistics.
- Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. [Www’18 open challenge: Financial opinion mining and question answering](#). *Companion Proceedings of the The Web Conference 2018*.
- P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65.
- Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. [Mteb: Massive text embedding benchmark](#). *arXiv preprint arXiv:2210.07316*, arXiv:2210.07316.
- Raj Shah, Kunal Chawla, Dheeraj Eidnani, Agam Shah, Wendi Du, Sudheer Chava, Natraj Raman, Charese Smiley, Jiaao Chen, and Diyi Yang. 2022. [When FLUE meets FLANG: Benchmarks and large pre-trained language model for financial domain](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2335, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#). *Preprint*, arXiv:2302.13971.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. [Representation learning with contrastive predictive coding](#). *Preprint*, arXiv:1807.03748.
- Liang Wang, Nan Yang, Xiaolong Huang, Bin-xing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024a. [Text embeddings by weakly-supervised contrastive pre-training](#). *Preprint*, arXiv:2212.03533.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. [Multilingual e5 text embeddings: A technical report](#). *Preprint*, arXiv:2402.05672.
- Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022a. [Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.
- Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. 2022b. [Robust fine-tuning of zero-shot models](#). *Preprint*, arXiv:2109.01903.
- Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhajan Kam-badur, David Rosenberg, and Gideon Mann. 2023. [Bloomberggpt: A large language model for finance](#). *Preprint*, arXiv:2303.17564.
- Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. [C-pack: Packaged resources to advance general chinese embedding](#). *Preprint*, arXiv:2309.07597.
- Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. [PIXIU: A large language model, instruction data and evaluation benchmark for finance](#). *Preprint*, arXiv:2306.05443.
- Yi Yang, Mark Christopher Siy UY, and Allen Huang. 2020. [Finbert: A pretrained language model for financial communications](#). *Preprint*, arXiv:2006.08097.



## A Appendix

### A.1 Instructions to Query Annotators

Your task is to read a text chunk and then write a query, such that the text chunk should be a top hit in a world-class retrieval system. Tips for writing queries:

1. The query should be closely related to some part of the text (but not necessarily the entire passage).
2. Queries can be from 1 to 15 words.
3. Queries can be a list of keywords, a full sentence, a question - whatever an analyst might search. Anything goes as long as the text chunk would be a very good search result for that query.
4. Diversity is important, don't make your queries all the same.
5. Broad queries about entire industries, sectors, regions, or macro trends are fine, as long as the text snippet contains specific information that is highly relevant to the query.
6. The query can mention a stock name or ticker. It doesn't have to.
7. Don't copy too many words and phrases directly from the text passage. Use paraphrasing, synonyms, summarization, and your knowledge of appropriate abbreviations, acronyms and specialized terminology to construct queries. E.g., if the text mentions 'SBC', the query might mention 'stock based comp'.

### A.2 LLM Prompt for Query Generation

You are a highly trained investment analyst, and an expert in business and financial markets. You are helping construct a dataset to train a world class financial search engine. You will be given a text snippet from <DOCUMENT\_TYPE>. Your task is to generate a query derived from the provided text snippet.

Detailed Instructions:

1. The query should be a question, or a set of keywords or phrases, such that the text snippet should be returned as a top search result for that query.
2. A good query is closely related to at least some, but not necessarily all, of the content in the text snippet. Do not create queries containing many unrelated concepts.
3. Broad queries about entire industries, sectors, regions, or macro trends are okay, as long as the text snippet contains specific information

that is relevant to the query.

4. You will be penalized if your query contains too many words and phrases copied directly from the text snippet. Use paraphrasing, synonyms, summarization, and your knowledge of appropriate abbreviations, acronyms and specialized terminology to construct queries. For example, if the text snippet contains the phrase "earnings per share", the query could instead include the acronym "EPS". If the text snippet mentions "earnings guidance for 2020-2024", the query could be for "long-term profit outlook".
5. Do not include 10k or 10q references in your queries.

Formatting Instructions:

1. Always reply with the query only, on a single line. Do not provide any additional context, note, or explanation of any kind. Do not put the query in quotation marks (",'). Do not include html tags in the query.
2. Queries can contain a maximum of 10 words.
3. If the text snippet is very short, difficult to understand, not written in English, or if it contains only boilerplate investment risk disclosures or disclaimers, you must begin your response with the special output "SKIP".

Text Snippet: <EXAMPLE\_PASSAGE\_1>

Query: <EXAMPLE\_QUERY\_1>

Text Snippet: <EXAMPLE\_PASSAGE\_2>

Query: <EXAMPLE\_QUERY\_2>

Text Snippet: <SAMPLED\_PASSAGE>

Query: