Advancing Semantic Caching for LLMs with Domain-Specific Embeddings and Synthetic Data

Waris Gill[1,2] **Justin Cechmanek**[1] **Tyler Hutcherson**[1] **Srijith Rajamohan**[1]
Jen Agarwal[1] **Muhammad Ali Gulzar**[2] **Manvinder Singh**[1] **Benoit Dion**[1]
[1]Redis, USA [2]Virginia Tech
{waris.gill, justin.cechmanek, tyler.hutcherson, srijith.rajamohan,
jen.agarwal, manvinder.singh, benoit.dion}@redis.com
waris@vt.edu, gulzar@cs.vt.edu

Abstract

This report investigates enhancing semantic caching effectiveness by employing specialized, fine-tuned embedding models. Semantic caching relies on embedding similarity rather than exact key matching, presenting unique challenges in balancing precision, query latency, and computational efficiency. We propose leveraging smaller, domain-specific embedding models, fine-tuned with targeted real-world and synthetically generated datasets. Our empirical evaluations demonstrate that compact embedding models fine-tuned for just one epoch on specialized datasets significantly surpass both state-of-the-art open-source and proprietary alternatives in precision and recall. Moreover, we introduce a novel synthetic data generation pipeline for the semantic cache that mitigates the challenge of limited domain-specific annotated data, further boosting embedding performance. Our approach effectively balances computational overhead and accuracy, establishing a viable and efficient strategy for practical semantic caching implementations.

1 Introduction

Large language models (LLMs) are rapidly becoming integral components of modern applications, significantly influencing everyday tasks. These models consist of billions of neurons and require substantial computational infrastructure to operate effectively. Each user query to an LLM involves billions of floating-point operations to generate an appropriate response Frantar et al. [2023].

In practice, users often issue repeated queries to online services. Prior research Lempel and Moran [2003], Xie and O'Hallaron [2002], Markatos [2001] indicates that approximately 33% of queries submitted to web search engines are repeated. A similar phenomenon occurs with LLM-based services Gill et al. [2025]. Recognizing this, researchers have proposed implementing a *semantic cache* to efficiently handle duplicate queries directed at LLMs Bang [2023], Gill et al. [2025], Zhu et al. [2023]. Unlike traditional key-value caches, a semantic cache does not rely on exact key matching. Instead, it declares a *cache hit* if the similarity between the embeddings generated from two textual inputs surpass a predefined cosine similarity threshold Bang [2023], Gill et al. [2025].

A semantic cache generally comprises two essential components: an embedding model, which computes query embeddings, and a vector database, which stores these embeddings and retrieves cached responses by matching embeddings of repeated queries. A variety of embedding models are available, ranging from open-source to proprietary closed-source models, both demonstrating state-of-the-art (SOTA) performance on benchmarks such as MTEB Muennighoff et al. [2023]. An important consideration thus arises: which embedding model is optimal for semantic caching? Both

Preprint. Under review.

open-source and closed-source options have their respective advantages and disadvantages. Open-source models are freely accessible, allowing users to maintain data privacy by running models on their own infrastructure. However, these models often contain billions of parameters, as exemplified by Alibaba-NLP/gte-Qwen2-7B-instruct (a 7-billion-parameter embedding model Li et al. [2023], Ali), making them computationally intensive and less suitable for efficient semantic caching. Conversely, closed-source embedding models offered via managed services are costly, may raise data privacy concerns due to external data handling, and introduce network latency, potentially hindering the performance of semantic caches.

Ideally, an embedding model used in semantic caching should be lightweight and computationally efficient. Unfortunately, smaller models typically lag behind their larger counterparts in performance. Nevertheless, recent studies have demonstrated that smaller, task-specific fine-tuned models can surpass the performance of significantly larger models Ouyang et al. [2022], Penedo et al. [2024], Du and Kaelbling [2024]. Motivated by this insight, we fine-tuned smaller embedding models on domain-specific datasets (medical and Quora kag [b] datasets). Remarkably, our experiments revealed that fine-tuning for just one epoch enabled these compact models to outperform both state-of-the-art closed-source and open-source models.

However, fine-tuning these smaller models requires high-quality datasets, which may not always be readily available. To address this limitation, we developed a unique synthetic data generation pipeline for the semantic cache. This pipeline leverages existing datasets from diverse domains to produce targeted synthetic data tailored specifically for different applications and domains. Our evaluation demonstrates that synthetic data significantly enhances the smaller embedding model performance for semantic caching, achieving precision improvements of 9% compared to its non-finetuned base model. Moreover, when tested on real-world medical queries, the model fine-tuned on medical synthetic data achieves performance approaching or even matching that of leading open-source and closed-source models, surpassing OpenAI's embedding model by 2% in precision.

2 Methodology

Embedding models lie at the heart of semantic caching. Their role is to produce high-quality, high-dimensional representations that encapsulate the semantic information in text. The better these models are at distinguishing subtle nuances in meaning, the more effective the cache becomes at reusing relevant results. State-of-the-art embedding models, whether open-source (such as Alibaba-NLP/gte-Qwen2-7B-instruct) Li et al. [2023], Ali, Zhang et al. [2024] or closed-source Ama, Emb, Vec, Lee et al. [2024, 2025], Neelakantan et al. [2022] , have shown remarkable effectiveness in tasks like sentence similarity, text classification, and semantic retrieval. In semantic caching, however, the overarching challenge is balancing performance with computational and operational constraints. While larger models often excel in capturing fine-grained nuances, they are costly to run repeatedly at scale. Conversely, more efficient, smaller models may falter on challenging queries, leading to suboptimal retrieval and degraded cache performance.

A key shortcoming of baseline (out-of-the-box) embedding models is that they are typically trained on broad, general-purpose corpora. When faced with domain-specific queries (e.g., finance, medical, or legal), these models may struggle to capture the nuanced relationships unique to that domain. This leads to lower performance, especially for duplicate queries that differ slightly in wording but share the same semantic content. For instance, in the medical domain, the queries "myocardial infarction treatment" and "how to treat a heart attack" may not be recognized as semantically equivalent by a general-purpose embedding model. Such limitations mean that, in practice, relying solely on these larger, general-purpose embedding models can either incur excessive compute overhead or fail to retrieve sufficiently accurate cached results for highly specialized queries. Consequently, there is a growing need for fine-tuned, domain-adapted embedding models, particularly compact ones, that can provide robust semantic representations while remaining efficient to deploy in production. To address the challenges outlined above, our methodology comprises three main components: (1) Model Selection and Training, (2) Domain-Specific Fine-Tuning, and (3) Synthetic Data Generation.

We begin by selecting a compact, publicly available embedding model, ModernBERT Warner et al. [2024], which contains approximately 149 million parameters. ModernBERT is a recent encoder-only transformer that outperforms comparable models such as BERT Devlin et al. [2019], NomicBERT Nussbaum et al. [2025], and RoBERTa Liu et al. [2019]. While our insights are broadly

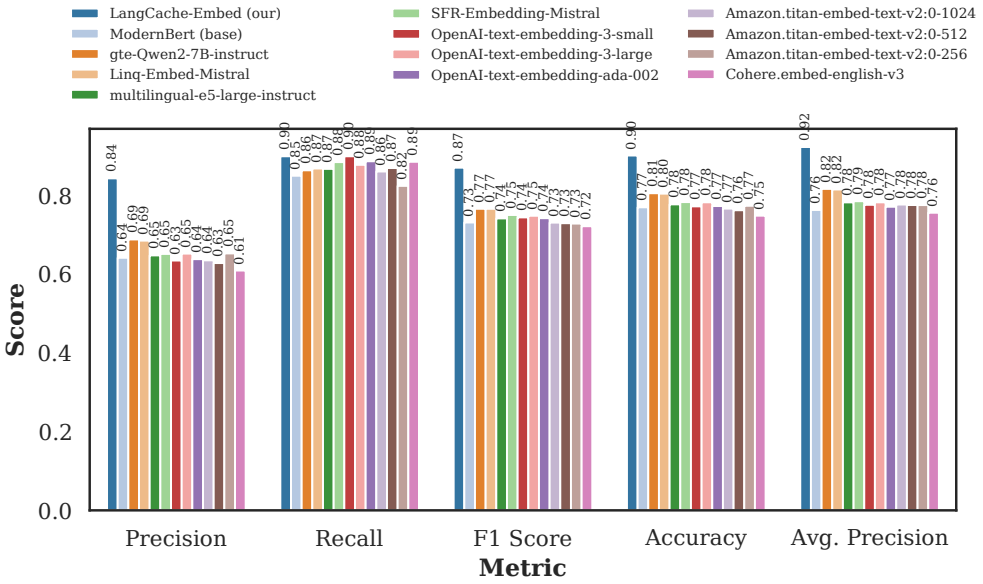

Figure 1: Comparison of embedding-model performance on the Quora dataset. The y-axis shows score, while the x-axis lists metrics. LangCache-Embed (i.e., fine-tuned ModernBERT) exhibits a significant uplift in precision and recall compared to its baseline (non-fine-tuned) version and other state-of-the-art embedding models, highlighting the impact of fine-tuning.

applicable to other smaller models like MiniLM Wang et al. [2020], MPNet Song et al. [2020], and ALBERT Lan et al. [2020], we choose ModernBERT for its strong balance of efficiency and performance. Models like ModernBERT are particularly advantageous for semantic caching due to their reduced computational footprint and faster inference times.

To fine-tune the ModernBERT embeddings for domain-specific queries, we use the online contrastive loss function Reimers and Gurevych [2019]. For simplicity and brevity, we refer to the fine-tuned ModernBERT model as *LangCache-Embed* in our evaluations, in contrast to the non-fine-tuned base ModernBERT hug, Zhang et al. [2024], Li et al. [2023]. Contrastive objectives encourage the model to produce similar vector representations for semantically similar inputs (duplicate queries) while pushing apart the representations of dissimilar inputs (distinct queries). For example, "reset my password" and "forgot login credentials" should be close in the embedding space, whereas "update billing info" should be far from them. However, a key limitation of conventional contrastive learning is that it treats all positive and negative examples equally. In contrast, online contrastive learning Reimers and Gurevych [2019], sbe examines an entire batch and focuses on the "hardest" examples, namely, positive pairs that the model currently ranks as relatively distant in embedding space, and negative pairs that the model ranks as relatively similar. By only computing the contrastive objective on these difficult pairs, the online contrastive loss accelerates learning in precisely the regions of the embedding space where the model is most likely to confuse positives and negatives. This ensures that the final model is more discriminative, which is crucial for detecting subtle differences between duplicate domain-specific queries. By focusing training on challenging examples, we observe faster convergence and better precision, which is especially beneficial in semantic caching Gill et al. [2025], where correctly recognizing duplicate queries is critical for true cache hit rates.

2.1 Synthetic Data Generation

A central challenge in developing domain-specific semantic caches is obtaining sufficient quantities of high-quality labeled data that accurately reflect the subtle ways in which users may pose similar or closely related queries. This issue is especially prominent in specialized domains such as medicine, where large and meticulously annotated datasets are often scarce. To overcome this limitation, we designed a synthetic data generation pipeline tailored to produce both positive (paraphrased)

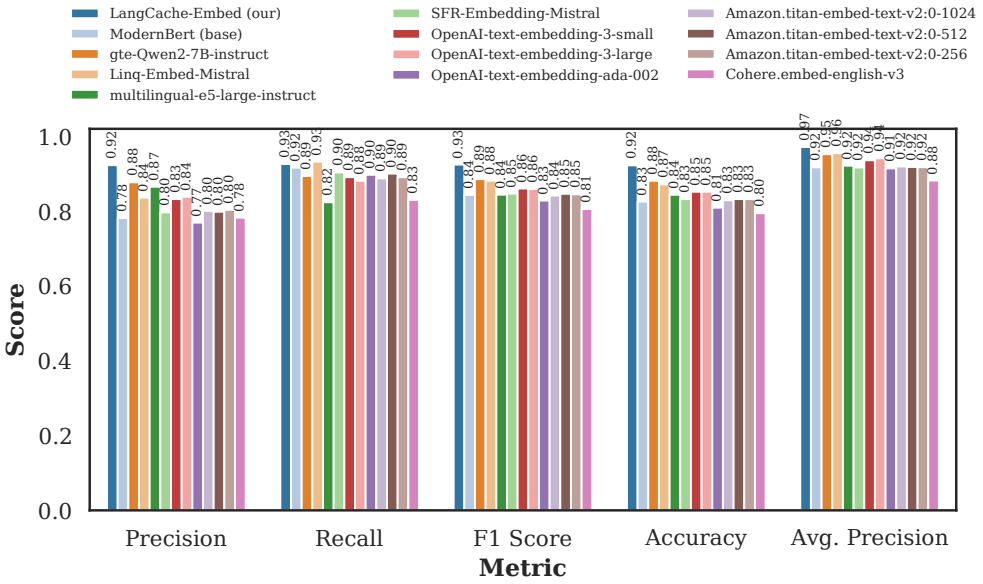

Figure 2: Evaluation of different embedding models on a specialized medical dataset. Notably, LangCache-Embed outperforms both large-scale open-source and closed-source baselines, demonstrating that lightweight models adapted to domain-specific data can achieve state-of-the-art results.

and negative (semantically related yet distinct) query pairs. This pipeline facilitates fine-tuning embedding models to more effectively distinguish near-duplicate queries from those merely related by topic.

Our synthetic data generation approach leverages a large language model (LLM). Recognizing the frequent absence of domain-specific labeled datasets suitable for semantic caching, we capitalize instead on the widespread availability of unlabeled query datasets within the target domain (e.g., medical queries to LLMs from open-source repositories). For each original query from these datasets, carefully structured prompts guide the LLM in generating two distinct types of synthetic variants. Firstly, we create **positive samples**, which are paraphrased queries that retain the intent of the original but differ in wording or syntax. These positive samples enable the model to identify queries that, despite differences in wording, convey identical semantic meanings, thereby reducing false negatives, instances where the cache fails to recognize semantically identical queries. For example, given the unlabeled query "Q1: What are the symptoms of early-stage diabetes?", the LLM might generate a positive sample such as "Q2: How can I tell if someone has diabetes in its initial phase?" In this case, Q1 and Q2 are semantically identical and labeled as duplicates (i.e., a positive sample).

Secondly, we generate **negative samples**, which comprise queries that, while topically related, diverge sufficiently in focus or subdomain. These examples help embedding models identify clear semantic boundaries, thereby reducing false positives, situations in which merely related queries are incorrectly treated as near-duplicates. For instance, given the same original query "Q1: What are the symptoms of early-stage diabetes?", a negative sample might be "Q3: What are common health risks in children with type 1 diabetes?" Although both queries involve diabetes, Q1 focuses on general early-stage symptoms, while Q3 pertains to pediatric care and type 1 diabetes, making them semantically distinct.

Several distinctive features characterize our synthetic data generation approach. Notably, we employ a dual-labeling strategy that simultaneously produces both paraphrased ($is\_duplicate = 1$) and distinct ($is\_duplicate = 0$) queries within the same pipeline. This strategy effectively captures semantic edge-cases encountered in real-world applications, where user queries partially overlap but warrant distinct responses. The synthetic data generated through our pipeline plays a crucial role in fine-tuning embedding models tailored specifically for semantic caching. By exposing these models to diverse paraphrased and closely related negative examples, we enhance their capacity to detect nuanced differences in user intent, domain-specific terminologies, and clinical contexts,

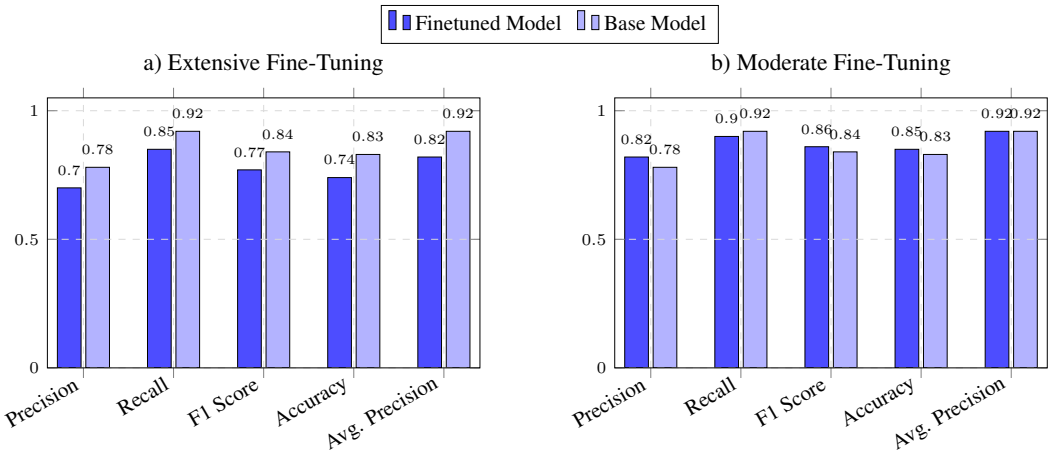

Figure 3: The plots compare performance on the target (fine-tuning) dataset versus performance on an unseen or previously learned dataset. a) Overly extensive fine-tuning on a single domain degrades the model's generalization to out-of-domain queries, by reduced precision. b) Limiting fine-tuning (e.g., to a single epoch and moderate gradient norm) mitigates catastrophic forgetting and preserves strong cross-domain performance.

which are typically lacking in generic, pre-trained models. Consequently, this approach improves precision, by preventing confusion among semantically distinct yet related queries. Ultimately, our semantic cache achieves performance comparable to, and often surpassing, state-of-the-art closed-source and open-source embedding models on the test dataset (Section 3.3). Practically, this means organizations can attain robust, domain-specific query-handling performance without incurring the substantial computational and licensing costs typically associated with larger, resource-intensive embedding solutions.

3 Evaluation

Experimental Setup. We perform all experiments using an Amazon EC2 G6e instance ama, equipped with 48 parallel processes, 384 GB of system RAM, and four NVIDIA L40S GPUs, each featuring 48 GB of dedicated GPU memory. To evaluate the performance of *ModernBERT*, we fine-tune the model (i.e., LangCache-Embed) on two distinct datasets: the Quora dataset kag [b] and a specialized medical dataset kag [a]. Each dataset comprises data points structured as pairs (Question1, Question2), accompanied by a binary label *is_duplicate*, which is set to 1 if the questions are semantically similar (i.e., duplicates), and 0 otherwise. The Quora dataset kag [b] includes $323,491$ training samples and $53,486$ evaluation samples, while the medical dataset kag [a] consists of $2,438$ training samples and 610 evaluation samples. The medical dataset kag [a] introduces unique challenges, as it requires distinguishing subtle semantic variations in questions authored by 11 different medical professionals. For example, the question pair "Can doxycycline treat an ear infection?" and "What are the side effects of doxycycline?" is labeled 0, indicating they are not duplicates. In contrast, in the Quora dataset kag [b], the pair "How can I be a good geologist?" and "What should I do to be a great geologist?" is labeled as 1, reflecting semantic duplication. Both datasets are well-suited to train specialized embeddings models for semantic caching, where the objective is to retrieve answers from cached responses for semantically similar queries, thereby reducing the need for repeated forward passes through LLMs (e.g., ChatGPT).

We compare our fine-tuned model against top-performing embedding models from the MTEB benchmark Muennighoff et al. [2023], encompassing both open-source models including multilingual-e5-large-instruct Wang et al. [2024], gte-Qwen2-7B-instruct Li et al. [2023], Linq-Embed-Mistral Kim et al. [2024], and SFR-Embedding-Mistral Meng et al. [2024], as well as leading closed-source alternatives from OpenAI Vec, Amazon Ama, and Cohere Emb. This comparison enables a thorough understanding of how *LangCache-Embed* performs relative to state-of-the-art solutions under different domain constraints.

We conduct the fine-tuning process using the SBERT library and employ the online contrastive loss function. Our choice of hyperparameters, one training epoch, a learning rate of $6.5383156211679 \times 10^{-5}$, a batch size of 16, the Adam optimizer, and a gradient norm of 0.5, strikes a balance between domain-specific adaptation and the preservation of prior knowledge. Specifically, we opt for a single epoch and a relatively small gradient norm to mitigate catastrophic forgetting, as prolonged fine-tuning can diminish the model's broader capabilities. Standard performance metrics such as Precision, Recall, F1-score, Average Precision (AP), and Accuracy are measured on both datasets, allowing us to capture the accurate comparisons.

3.1 Domain-Specific Fine-Tuning Yields State-of-the-Art Performance

We observe that domain-specific fine-tuning consistently enhances *ModernBERT*'s performance on both Quora and medical datasets. For brevity and clear distinction, we refer to the finetuned ModernBERT as *LangCache-Embed* in the evaluations. Figure 1 and Figure 2 illustrate the improvements in key metrics. On the Quora dataset, *LangCache-Embed*'s average precision increases from 76% to 92%, while on the medical dataset, it improves from 92% to 97%. This substantial gain highlights that fine-tuning effectively aligns model representations with domain nuances. Precision, in particular, is often critical for tasks such as semantic caching Gill et al. [2025], which require accurate retrieval of the most relevant items. Hence, we note that fine-tuning on Quora elevates *LangCache-Embed*'s precision from 64% to 84%, while on the medical dataset it rises from 78% to 92%. These consistent improvements span multiple metrics and exemplify how specialized training data can drastically refine model embeddings.

When compared to top models from the MTEB benchmark, including both open-source and proprietary options, fine-tuned *ModernBERT* (i.e., LangCache-Embed) yields state-of-the-art performance on both the Quora and medical datasets. As depicted in Figure 1 and Figure 2, our model surpasses leading closed-source models and outperforms the best open-source solutions. For instance, Figure 2 shows that the LangCache-Embed achieves a 6% improvement over OpenAI's best embedding model (text-embedding-3-large), reinforcing that our approach maintains competitiveness even against well-established, proprietary embedding models.

3.2 Avoiding Catastrophic Forgetting with Controlled Fine-Tuning

Catastrophic forgetting is when a neural network forgets previously learned tasks after being trained on new ones Zhai et al. [2023], Franke et al. [2024]. It's a problem because it prevents models from learning continuously without losing past knowledge. This happens because neural networks update weights during training, often overwriting old information in the process. For example, if a neural network is trained to recognize animals, and then trained to recognize vehicles without revisiting the animal data, it might completely "forget" how to recognize animals. Although extended fine-tuning can yield superior performance in a single domain, it often erodes the model's generalization capabilities, as illustrated in Figure 3-a. For instance, after six epochs of fine-tuning on the Quora dataset, we observe a 22% improvement in precision compared to the non-trained base ModernBERT on Quora test data. However, fine-tuned ModernBERT on Quora shows an 8% drop in precision when evaluated on a medical test data, compared to its base (untrained) version, illustrating catastrophic forgetting (Figure 3-a). This degradation indicates that excessive updates to adapt to one domain can overshadow the model's prior knowledge. By contrast, limiting fine-tuning to a single epoch and constraining the gradient norm to 0.5 strikes a better balance in our experiments. As shown in Figure 3-b, when we adopt these settings, the model not only retains its prior knowledge but also improves by 4% in precision on the medical dataset after being trained on Quora dataset, thus underscoring the importance of moderate fine-tuning for preserving broader model competencies.

3.3 Synthetic Data for Domain Adaptation

```
You are a helpful medical expert. Generate 2 unique paraphrases of the given query.
Original Query: '{query}'
Each paraphrase should:
1. Preserve the original meaning but use different wording or sentence structure.
2. Avoid changing medical intent or introducing new information.
3. Be professionally written and clear.
Example:
```

Model	Source	Precision	Recall	F1	Accuracy	Avg. Precision
OpenAI-text-embedding-3-small	Closed	0.83	0.89	0.86	0.85	0.94
OpenAI-text-embedding-3-large	Closed	0.85	0.87	0.86	0.85	0.94
OpenAI-text-embedding-ada-002	Closed	0.77	0.90	0.83	0.81	0.91
Amazon.titan-embed-v2:0-1024	Closed	0.80	0.89	0.84	0.83	0.92
Amazon.titan-embed-v2:0-512	Closed	0.84	0.86	0.85	0.84	0.92
Amazon.titan-embed-v2:0-256	Closed	0.80	0.89	0.85	0.83	0.92
Cohere.embed-english-v3	Closed	0.78	0.83	0.81	0.80	0.88
Linq-Embed-Mistral	Open	0.84	0.93	0.88	0.87	0.96
multilingual-e5-large-instruct	Open	**0.87**	0.82	0.84	0.84	0.92
SFR-Embedding-Mistral	Open	0.80	0.90	0.85	0.83	0.92
gte-modernbert-base	Open	0.78	0.89	0.84	0.83	0.92
LangCache-Embed-Synthetic	Open	**0.87**	0.90	0.89	0.88	0.95

Table 1: Effect of fine-tuning ModernBERT on a purely **synthetic medical dataset** (**LangCache-Embed-Synthetic**) and then evaluating on the real-world medical dataset kag [a]. Results show that leveraging synthetic dataset significantly boosts in-domain performance, allowing ModernBERT (**LangCache-Embed-Synthetic**) to rival or surpass larger closed-source models in both precision and recall.

```
Original Query: "What are the best ways to reduce stress?"
Paraphrased Queries:
1. ‘‘How can a person effectively manage stress?"
2. ‘‘What strategies help in reducing stress levels?"
Return JSON with a key ‘queries’ containing a list of the two paraphrased versions.
```

Listing 1: Prompt used for paraphrase generation.

```
You are a helpful medical expert. Given a medical query, generate two distinct but
    related queries that explore different aspects of the topic.
Guidelines:
1. The new queries should be related to the original but focus on different subtopics,
    perspectives, or medical contexts.
2. They should not be simple rewordings or slight variations of the original.
3. Consider different patient populations, alternative diagnostic methods, treatments, or
     physiological explanations.
Examples:
Original Query:
"How to reduce stress?"
 Distinct Queries:
1. ‘‘How can athletes manage stress during high-pressure competitions?" (Context: Sports
    Psychology)
2. ‘‘What are effective stress management strategies for children with ADHD?" (Context:
    Pediatric Stress Management)
Original Query:
‘‘A 61-year-old woman with a long history of involuntary urine loss during activities
    like coughing or sneezing but no leakage at night undergoes a gynecological exam and
     Q-tip test. Based on these findings, what would cystometry most likely reveal about
     her residual volume and detrusor contractions?"
Distinct Queries:
1. ‘‘How does the Q-tip test help differentiate between stress urinary incontinence and
    urge incontinence?" (Context: Diagnostic Techniques)
2. ‘‘What are the treatment options for stress urinary incontinence in postmenopausal
    women, and how does cystometry aid in management?" (Context: Treatment & Management)

Now, generate two distinct queries for this input:
Original Query: {query}
Return JSON with ‘queries’ only.
```

Listing 2: Prompt used for non-duplicate (distinct) queries generation.

To address 1 the limited availability of annotated data in specialized domains (e.g., medical), we introduce a synthetic data generation pipeline designed to enhance domain adaptation without incurring the high costs of human annotation. Recent work by Chen et al. [2024] has made approximately 25,000 medical queries, along with corresponding chain-of-thought reasoning and responses, publicly accessible. Leveraging this resource, we use Qwen2.5 with 32 billions parameters Yang et al.

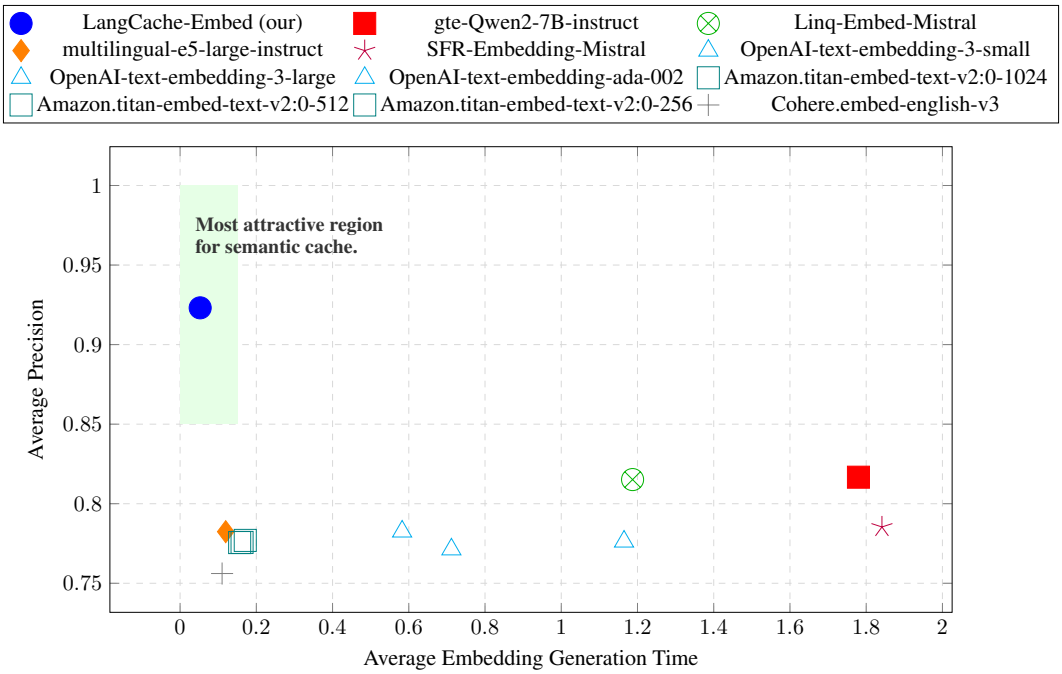

Figure 4: This plot illustrates the trade-off between embedding generation overhead (x-axis, measured in seconds) and average precision on the Quora test set (y-axis). Each point represents a different embedding model, including both open-source and commercial offerings. Models in the upper-left region deliver high precision at low embedding time. LangCache-Embed (finetuned ModernBERT) stands out for combining rapid inference with top-tier performance, indicating it is an ideal choice for real-time semantic caching where both speed and accuracy are critical.

[2024] to generate both semantically similar and dissimilar query pairs (Section 2.1), using prompts shown in Listings 1 and 2, respectively. This results in approximately 35,000 synthetic samples relevant to medical semantic task.

We fine-tune *ModernBERT* (LangCache-Embed) on this synthetic dataset and then evaluate it on a real-world medical dataset kag [a], observing a marked improvement. As shown in Table 1, the precision of *LangCache-Embed* increases from 78% to 87%, showcasing a 9% gain through purely synthetic data. Crucially, these results demonstrate that domain-specific knowledge can be effectively injected into the model by training on synthetic pairs that approximate realistic in-domain relationships for the purpose of embedding generation. Moreover, LangCache-Embed now matches the performance of the best OpenAI embedding model and even exceeds it by 2% in terms of precision, while surpassing Cohere's closed-source embedding model by 9% (Table 1). This highlights the effectiveness of synthetic data generation in bridging data gaps in semantic caching, allowing models to excel in specialized domains without direct reliance on large-scale human-annotated queries, as described in Section 2.1.

3.4 Embedding Generation Overhead (Latency)

Embedding generation overhead is a critical consideration for semantic caching, as it should not disrupt the overall flow of the generative AI application. Some approaches, such as using LLMs as the embedding model (e.g., Llama) to enhance semantic caching Bang [2023], are regarded as impractical due to the high computational and memory demands Gill et al. [2025]. We therefore evaluate the embedding generation overhead across a range of open-source and proprietary models, incorporating not just local computation times but also the latency of any external API calls in the case of closed-source services.

Figure 4 illustrates the trade-off between embedding generation time (on the X-axis) and average precision on Quora test data (on the Y-axis). These measurements are obtained using CPUs rather than GPUs, as many semantic caching environments may lack access to specialized hardware. Notably, experiments on an Amazon EC2 instance reveal that Amazon Titan and Cohere exhibit comparatively lower API latencies, particularly when deployed in the same AWS region, thereby reducing network overhead.

We observe that fine-tuned *ModernBERT* (LangCache-Embed) delivers the lowest embedding generation overhead while achieving superior performance and thus emerges as an optimal choice, striking a balance between rapid embedding generation and high precision. In cases where organizational policies mandate the use of proprietary solutions, Amazon Titan also proves effective within the AWS ecosystem, offering low latency and robust performance when accessed through Bedrock APIs.

4 Conclusion

In summary, our work demonstrates that smaller embedding models, when carefully fine-tuned on domain-specific or synthetic data, can outperform significantly larger open-source and closed-source models for semantic caching. By restricting fine-tuning and carefully managing gradient norms, we avoid catastrophic forgetting and preserve general performance. Moreover, our synthetic data generation pipeline effectively addresses the scarcity of annotated domain data for semantic caching. Together, these techniques enable a lightweight, high-performing embedding model that combines efficiency with high average precision, providing a practical alternative to resource-intensive, large-scale models for semantic caching.

References

Alibaba-NLP/gte-Qwen2-7B-instruct · Hugging Face — huggingface.co. https://huggingface.co/Alibaba-NLP/gte-Qwen2-7B-instruct. [Accessed 01-04-2025].

Amazon Titan Text Embeddings models - Amazon Bedrock — docs.aws.amazon.com. https://docs.aws.amazon.com/bedrock/latest/userguide/titan-embedding-models.html. [Accessed 01-04-2025].

Embed: The Leading Text Representation Language Model — cohere.com. https://cohere.com/embed. [Accessed 01-04-2025].

Vector embeddings. https://platform.openai.com/docs/guides/embeddings. [Accessed 01-04-2025].

Amazon EC2 G6e Instances — Amazon Web Services — aws.amazon.com. https://aws.amazon.com/ec2/instance-types/g6e/. [Accessed 01-04-2025].

Alibaba-NLP/gte-modernbert-base · Hugging Face — huggingface.co. https://huggingface.co/Alibaba-NLP/gte-modernbert-base. [Accessed 03-04-2025].

Medical Question Pair — kaggle.com. https://www.kaggle.com/datasets/thedevastator/medical-question-pair-classification/data, a. [Accessed 31-03-2025].

Question Pairs Dataset — kaggle.com. https://www.kaggle.com/datasets/quora/question-pairs-dataset, b. [Accessed 31-03-2025].

Losses: Sentence Transformers documentation — sbert.net. https://www.sbert.net/docs/package_reference/sentence_transformer/losses.html. [Accessed 31-03-2025].

Fu Bang. Gptcache: An open-source semantic cache for llm applications enabling faster answers and cost savings. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 212–218, 2023.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. Huatuogpt-o1, towards medical complex reasoning with llms, 2024. URL https://arxiv.org/abs/2412.18925.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Yilun Du and Leslie Kaelbling. Compositional generative modeling: A single model is not all you need. *arXiv preprint arXiv:2402.01103*, 2024.

JK Franke, Michael Hefenbrock, and Frank Hutter. Preserving principal subspaces to reduce catastrophic forgetting in fine-tuning. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=tcbBPnfwxS.

Waris Gill, Mohamed Elidrisi, Pallavi Kalapatapu, Ammar Ahmed, Ali Anwar, and Muhammad Ali Gulzar. MeanCache: User-Centric Semantic Caching for LLM Web Services. In *2025 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2025. URL https://arxiv.org/abs/2403.02694.

Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Yejin Kim, Minkyung Cho, Jy yong Sohn, and Chanyeol Choi. Linq-embed-mistral:elevating text retrieval with improved gpt data through task-specific control and quality refinement. Linq AI Research Blog, 2024. URL https://getlinq.com/blog/linq-embed-mistral/.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1eA7AEtvS.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, et al. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327*, 2024.

Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftekhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, et al. Gemini embedding: Generalizable embeddings from gemini. *arXiv preprint arXiv:2503.07891*, 2025.

Ronny Lempel and Shlomo Moran. Predictive caching and prefetching of query results in search engines. In *Proceedings of the 12th international conference on World Wide Web*, pages 19–28, 2003.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Evangelos P. Markatos. On caching search engine query results. *Computer Communications*, 24(2): 137–143, 2001.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog, 2024. URL https://www.salesforce.com/blog/sfr-embedding/.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. MTEB: Massive text embedding benchmark. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.148. URL https://aclanthology.org/2023.eacl-main.148/.

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

Zach Nussbaum, John Xavier Morris, Andriy Mulyar, and Brandon Duderstadt. Nomic embed: Training a reproducible long context text embedder. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=IPmzyQSiQE. Reproducibility Certification.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data only. *Advances in Neural Information Processing Systems*, 36, 2024.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33: 16857–16867, 2020.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.

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, 2020.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*, 2024.

Yinglian Xie and David O'Hallaron. Locality in search engine queries and its implications for caching. In *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1238–1247. IEEE, 2002.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. Investigating the catastrophic forgetting in multimodal large language model fine-tuning. In *Conference on Parsimony and Learning (Proceedings Track)*, 2023. URL https://openreview.net/forum?id=g7rMSiNtmA.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, et al. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1393–1412, 2024.

Banghua Zhu, Ying Sheng, Lianmin Zheng, Clark Barrett, Michael Jordan, and Jiantao Jiao. Towards optimal caching and model selection for large model inference. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=gd20oaZqqF.