SQUARE: Unsupervised Retrieval Adaptation via Synthetic Data

Jinsung Yoon, Junhao Zeng, Sercan Ö. Arık
Google Cloud
{jinsungyoon, junhaozeng, soarik}@google.com

Abstract

Pre-trained retrieval models often face challenges in zero-shot retrieval for knowledge-based question answering, as different tasks rely on different corpora. We introduce SQUARE (Synthetic QUery-based Adaptive REtrieval), a novel method for corpus-specific unsupervised retrieval customization. SQUARE leverages LLMs to generate grounded synthetic question-answer pairs from the corpus, which are then used to fine-tune the retriever. A filtering mechanism based on the synthetic answers is employed to ensure high quality of tuning data. Extensive experiments on various datasets demonstrate superior performance of SQUARE compared to zero-shot retrieval and other customization methods, highlighting the value of corpus adaptation for effective retrieval.

1 Introduction

Large language models (LLMs) frequently use retrieval-augmented generation to improve their question-answering capabilities [10, 5, 16]. It is based on retrieving relevant information from a large text corpus, allowing LLMs to produce more accurate and comprehensive responses. Retrieval is essential for grounding LLM responses, ensuring consistency and relevance to the information within the corpus [23, 7]. The accuracy of the retrieval process often directly determines the quality of the LLMs' answers.

Pre-trained dense embedding models are widely used for retrieval and often demonstrate strong zero-shot performance across diverse corpora [11, 8, 9]. However, their effectiveness can decline when the target corpus significantly deviates from the training data [21]. To overcome this bottleneck, customizing the retrieval model has the potential to yield significant improvements. Existing customization methods often rely on labeled query-document pairs from the corpus [22, 18], employing techniques like adaptor approaches, parameter-efficient fine-tuning [6], and full model fine-tuning. However, obtaining such labeled data can be costly and time-consuming. Moreover, some fine-tuning approaches often require substantial amounts of labeled data, limiting their practicality.

Unsupervised customization provides a more practical approach to customize retrieval, eliminating the need for costly labeled query-document pairs. This approach is designed to leverage the target corpus directly, which is readily available. Existing unsupervised techniques include Promptagator [3], which uses LLMs to generate relevant queries from passages (with filtering), and document expansion [14], where LLMs generate and append synthetic queries to each document. However, they have major bottlenecks. Promptagator's effectiveness is limited by its inability to handle queries related to multiple documents (due to its strict one-to-one query-document pairing) and its weak error filtering (relying solely on round-trip consistency with generated synthetic samples). Document expansion suffers from scalability issues, requiring synthetic query generation for the entire corpus to avoid retrieval bias.

To overcome these limitations, we introduce SQUARE (Synthetic QUery-based Adaptive REtrieval), a novel unsupervised retrieval customization method. SQUARE generates both synthetic queries and corresponding answers for each document of the corpus. These synthetic answers are then used to robustly filter noisy or irrelevant queries and identify additional relevant documents for each query. The resulting synthetic query-document pairs are used to fine-tune the retrieval model via supervised customization techniques.

We specifically selected newer retrieval benchmarks—BRIGHT [17], LongEmbed [24], and MTEB-Law [19]—from the MTEB leaderboard [13] to evaluate SQUARE. These benchmarks, not widely used in pre-training, provide an opportu-

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 7283–7297
November 4-9, 2025

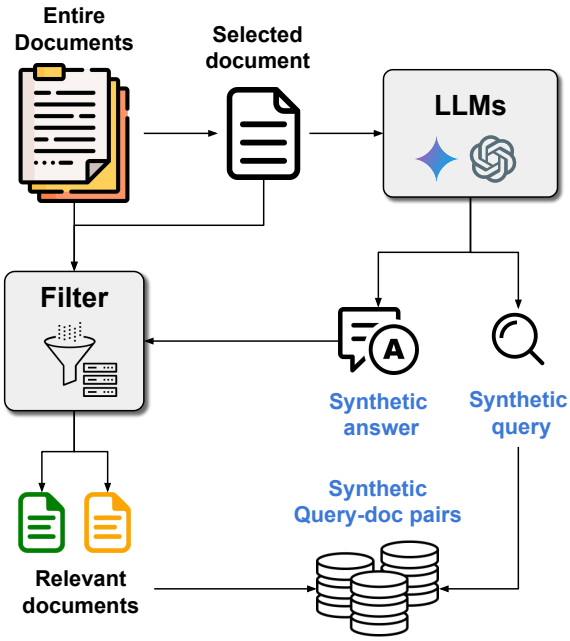

Figure 1: Overview of the SQUARE approach. From the document corpus, a single document is chosen and an LLM generates synthetic query-answer pairs, ensuring the answers are grounded to the selected document. These synthetic answers are then used to filter the remaining documents for relevance. The resulting relevant documents, along with the synthetic queries, form the synthetic data used to customize the retriever.

nity to measure how well our unsupervised customization generalizes to unseen data. Our results demonstrate significant performance gains compared to the zero-shot baselines and other unsupervised tuning methods, showcasing the effectiveness of SQUARE in improving retrieval accuracy for various LLM applications.

2 Method

In this section, we present SQUARE, a novel unsupervised framework designed to enhance retrieval performance. The complete workflow of SQUARE is described in Fig. 1. Given a corpus $\mathcal{D} = \{d_1, d_2, ..., d_N\}$ containing N documents, our goal is to extract information to generate synthetic data for customizing a retrieval model, ultimately improving retrieval accuracy within $\mathcal{D}$.

We leverage an LLM which inputs and produces text. For each document d_k, we prompt LLM with an instruction prompt I_p to generate a synthetic query $\hat{q}_k$ and its corresponding answer $\hat{a}_k$:

$$\hat{q}_k, \hat{a}_k = \text{LLM}(I_p, d_k). \quad (1)$$

Details of the prompting strategy are provided in Appendix A.

The generated synthetic query-answer-document triplets are represented as $\hat{\mathcal{D}} = \{(\hat{q}_i, \hat{a}_i, d_i)\}_{i=1}^{N}$. Unlike Document Expansion [14], SQUARE doesn't necessitate generating synthetic data for every document. For large corpora, generating data for a representative subset is sufficient, significantly enhancing scalability. Our experiments confirm that adapting with a subset of documents effectively boosts retrieval performance.

The generated answers $\hat{a}_k$ are crucial for refining the relationship between the synthetic query $\hat{q}_k$ and the documents in $\mathcal{D}$. We employ a pre-trained embedding model E that transforms text into numerical vectors, enabling similarity comparisons. We compute the similarity between the embedding of the generated answer, $E(\hat{a}_k)$, and the embedding of each document d_i in the corpus, $E(d_i)$:

$$s(\hat{a}_k, d_i) = \text{Sim}(E(\hat{a}_k), E(d_i)). \quad (2)$$

If the similarity score between $E(\hat{a}_k)$ and $E(d_k)$ is higher than the threshold $\eta_K(\hat{a}_k, \mathcal{D})$, we consider $(\hat{q}_k, d_k)$ a relevant query-document pair and add it to our filtered set:

$$\tilde{\mathcal{D}}_o = \{(\hat{q}, \hat{a}, d) \in \hat{\mathcal{D}} | s(\hat{a}, d) > \eta_K(\hat{a}, \mathcal{D})\}, \quad (3)$$

where $\eta_K(\hat{a}, \mathcal{D})$ is top-K-th similarity scores among $s(\hat{a}, d)$ where d in the entire corpus. This filtering step is introduced to remove low-quality synthetic query-answer pairs, where the generated answer lacks strong grounding in the corresponding document.

We then expand this set by identifying other potentially relevant documents. If any document d_i has a higher similarity score with $E(\hat{a}_k)$ than the similarity between $E(\hat{a}_k)$ and $E(d_k)$, we also include $(\hat{q}_k, d_i)$ in our synthetic query-document pair set:

$$\tilde{\mathcal{D}}_f = \{(\hat{q}_k, \hat{a}_k, d) | s(\hat{a}_k, d) \geq s(\hat{a}_k, d_k) \text{ where } d \in \mathcal{D} \text{ and } (\hat{q}_k, \hat{a}_k, d_k) \in \tilde{\mathcal{D}}_o\}. \quad (4)$$

Note that for this stage, we use only the synthetic query-answer pairs that survived from the previous filtering process ($\tilde{\mathcal{D}}_o$).

These filtering and expansion methods offer two key advantages. First, they exploit the direct relationship between generated answers and source documents, leading to more robust relevance learning, especially beneficial in complex question-answering scenarios requiring reasoning and inference (e.g., BRIGHT benchmarks [17]). Second,

Datasets	Short documents				Long documents			
	Zero	Prompt	Doc	SQUARE	Zero	Prompt	Doc	SQUARE
Biology	0.4035	0.4294	0.4227	**0.4941**	0.4167	0.4213	0.4221	**0.4717**
Earth-Science	0.4377	0.4596	0.4246	**0.4931**	0.3843	0.3922	0.3914	**0.4274**
Economics	0.2673	0.2904	0.2674	**0.3233**	0.2087	0.2379	0.2476	**0.3091**
Psychology	0.3641	0.3848	0.3639	**0.4208**	0.3366	0.3628	0.3422	**0.4574**
Robotics	0.1746	0.1752	0.1845	**0.2091**	0.1188	0.1288	0.1240	**0.1584**
Stackoverflow	0.2716	0.2724	0.2722	**0.2969**	0.1923	0.2038	0.2017	**0.2350**
Sustainable living	0.1687	0.1931	0.1761	**0.2493**	0.3282	0.3692	0.3772	**0.3932**
Pony	**0.0259**	0.0199	0.0042	0.0218	0.0229	**0.0237**	0.0207	0.0041
Average	0.2642	0.2781	0.2645	**0.3136**	0.2511	0.2675	0.2659	**0.3070**

Table 1: Retrieval performance with BRIGHT datasets in nDCG@10 (for short document) and Recall@1 (for long document). Bold represents the best performance per each dataset.

they allow a single query to be associated with multiple relevant documents, addressing a limitation of previous methods [3] and reflecting the common scenario of diverse and heterogeneous corpora in modern information retrieval systems.

We now have a set of filtered synthetic query-answer-document triplets, denoted as $\tilde{\mathcal{D}}_f$. This synthetic dataset is then used to fine-tune the retriever model in a supervised fashion.

3 Experiments

3.1 Experimental settings

We evaluate the effectiveness of SQUARE on three retrieval benchmarks from the MTEB leaderboard: BRIGHT [17], LongEmbed [24], and MTEB-Law [19]. These relatively new additions to MTEB [13] are selected specifically as they haven't been extensively used in pre-trained embedding model training, allowing to assess the impact of our unsupervised customization on unseen data. Dataset details can be found in Appendix B. We use Gemini-1.5-Flash-002 as the LLM for generating synthetic query-answer pairs in Eq. 1. We also includes experimental results using Gemini-2.0-Flash-001 in Appendix D, highlighting the LLM-agnostic nature of SQUARE.

We compare SQUARE against three baselines: (i) Zero-shot (A pre-trained embedding model without any customization), (ii) Promptagator (fully unsupervised version) [3], (iii) Document Expansion [14]. Our evaluation of retrieval quality across the three benchmarks centers on nDCG@10, a metric that effectively captures the relevance and ranking of the top 10 retrieved documents. Nevertheless, to ensure alignment with the established evaluation protocols of the BRIGHT benchmark, particularly for the BRIGHT-Long document dataset, we utilize Recall@1. See Appendix C for the additional experimental details.

Long-Embed datasets				
Sub-data	Zero	Prompt	Doc	SQUARE
Wikim-QA	0.7288	0.7371	0.7482	**0.7642**
QMSum	0.3830	0.3901	0.3936	**0.4824**
Summ screen FD	0.9328	0.9336	0.9505	**0.9542**
Narrative-QA	0.3267	0.3695	0.3441	**0.4050**
Average	0.5928	0.6076	0.6091	**0.6515**
MTEB-Law datasets				
AILA-casedocs	0.3416	0.3530	0.3308	**0.3691**
AILA-statutes	0.4104	0.4178	0.4180	**0.4282**
Consumer contracts	0.8301	0.8348	0.8410	**0.8393**
Corporate lobbying	0.9331	0.9284	0.9321	**0.9466**
Legal summary	0.6266	0.6340	0.6587	**0.6768**
Average	0.6284	0.6336	0.6361	**0.6520**

Table 2: Retrieval performance with Long-Embed and MTEB-Law datasets in terms of nDCG@10.

3.2 Experimental results

We fine-tune Google's Gecko-004 dense retriever using a Search-Adaptor [22] and our generated synthetic query-document pairs. While we employ this specific method, SQUARE is compatible with other fine-tuning techniques, which we explore in Sec. 4.3. Furthermore, despite using the Gecko-004 for these experiments, SQUARE is retriever-agnostic, as demonstrated by results with alternative retrievers presented in Sec. 4.1.

Our results, starting with Table 1, demonstrate that SQUARE significantly outperforms the zero-shot baseline and other unsupervised tuning methods across all BRIGHT datasets consistently, encompassing both long and short document settings. This substantial improvement highlights the effectiveness of our approach, especially given that it requires no labeled data. Furthermore, as shown in Table 2, SQUARE achieves similar performance

gains on LongEmbed and MTEB-Law, demonstrating its generalizability across diverse datasets. Notably, we use the same prompt (detailed in Appendix A) for all experiments, underscoring the robustness of our method which is critical in unsupervised settings. Some qualitative analyses can be found in Appendix F.

4 Discussions

4.1 Combine with other dense retrievers

To gain a deeper understanding of SQUARE's effectiveness, we conduct a series of analyses. We first investigate how well SQUARE generalizes across different retriever models. In addition to Google's Gecko-004, we also consider OpenAI-embedding-3-small and GTE-Large [11].

Retriever models	Method	BRIGHT		Long	MTEB
		Short	Long	Embed	-Law
Google-Gecko	Zero	0.2642	0.2511	0.5928	0.6284
	SQUARE	**0.3136**	**0.3070**	**0.6515**	**0.6520**
OpenAI-3-small	Zero	0.2634	0.2229	0.5234	0.6124
	SQUARE	**0.2877**	**0.2761**	**0.5995**	**0.6242**
GTE-Large	Zero	0.1867	0.2285	0.5661	0.4176
	SQUARE	**0.2044**	**0.2767**	**0.5667**	**0.4871**

Table 3: Performance of SQUARE across different retriever models. Full results are in Appendix E.

We apply SQUARE to these three models (with Search-Adaptor being the fine-tuning approach). The performance gains shown in Table 3 highlight SQUARE's effectiveness in enhancing a range of retriever models through unsupervised adaptation.

4.2 Ablation study

To pinpoint the specific factors contributing to the performance gains, we perform ablation studies, systematically removing critical components of SQUARE and observing the resulting performance degradation. By analyzing the impact of each component's absence, we can discern the individual contributions of components.

The ablation study presented in Table 4 elucidates the individual impact of each component on performance, offering a detailed understanding of the method's effectiveness and confirming their corresponding importances. SQUARE's effectiveness is notably enhanced by its answer-based filtering and its capacity to identify multiple relevant documents for each query.

Datasets	Zero	SQUARE	w/o filtering	w/o multi relevance
Biology	0.4035	**0.4941**	0.4803	0.4670
Earth Science	0.4377	**0.4931**	0.4735	0.4848
Economics	0.2673	**0.3233**	0.3146	0.3039
Psychology	0.3641	**0.4208**	0.4077	0.4101
Robotics	0.1746	**0.2091**	0.2025	0.1966
Stackoverflow	0.2716	**0.2969**	0.2778	0.2815
Sustainable living	0.1687	**0.2493**	0.2170	0.2209
Pony	**0.0259**	0.0218	0.0075	0.0105
Average	0.2642	**0.3136**	0.2976	0.2969

Table 4: Ablation studies for SQUARE. Without filtering represents skipping Eq. 3. Without multi rel. represents skipping Eq. 4.

4.3 Combine with other fine-tuning methods

We explore different fine-tuning approaches with the synthetic datasets generated by SQUARE. Alongside Search-Adaptor, we employ LoRA [6] and OpenAI cookbook[1] as alternative fine-tuning approaches using the generated synthetic query-corpus pairs. It is important to note that LoRA requires access to model weights (beyond just prediction API-based access) – thus, we use GTE-Large [11] as the dense retriever and apply the three approaches on it with synthetic data generated by SQUARE.

Datasets	Zero	Search -Adaptor	LoRA	OpenAI Cookbook
BRIGHT-Short	0.1867	**0.2044**	0.1861	0.1866
BRIGHT-Long	0.2285	**0.2767**	0.2301	0.2243
Long-Embed	0.5661	0.5667	0.5709	**0.5731**
MTEB-Law	0.4176	**0.4871**	0.3387	0.4364

Table 5: Performance of the proposed method across different fine-tuning method. Here, we use GTE-Large as the retriever method because this can be applicable for all three fine-tuning methods. Full results can be found in Appendix E.

Table 5 shows that only Search-Adaptor consistently improves performance over the zero-shot baseline. This consistency likely stems from Search-Adaptor's suitability for data-scarce scenarios. While LoRA and the OpenAI cookbook achieve clear improvements on some datasets (full results in Appendix E), they lack the robustness of Search-Adaptor across all datasets. This is a consistent pattern with the Search-Adaptor findings, even when they used the original, unaltered query-document pairs.

[1] https://cookbook.openai.com/examples/customizing_embeddings

4.4 SQUARE with various base LLMs

To ensure SQUARE works with different LLMs, we test it using Claude 3.5 Sonnet v2[2] to create synthetic query-answer pairs, in addition to our previous experiments with Gemini-1.5-Flash-002. We then use Search-Adaptor to fine-tune Google's Gecko-004 embedding models with the SQUARE-generated and filtered synthetic data. As shown in Table 6, various base LLM models yielded high-quality synthetic queries, leading to improved retrieval performance compared to the zero-shot baseline. This validates SQUARE's generalizability across diverse LLMs and more results can be found in Appendix D.

Datasets	Zero	SQUARE	
		Gemini-1.5 -Flash	Claude-3.5 -Sonnet-v2
Wikim-QA	0.7288	0.7642	0.7509
QMSum	0.3830	0.4824	0.4686
Summ screen FD	0.9328	0.9542	0.9529
Narrative-QA	0.3267	0.4050	0.4075
Average	0.5928	0.6515	0.6450

Table 6: Retrieval performance with LongEmbed datasets in nDCG@10 using two base LLM models.

5 Related Works

SQUARE is distinct from HyDE [4], which generates hypothetical documents from the query at inference time and averages their embeddings. This not only incurs a significant computational cost during inference—requiring multiple LLM invocations and embedding calculations per query—but also differs fundamentally from SQUARE. SQUARE generates synthetic queries and answers directly from the documents during the tuning phase, ensuring all knowledge is grounded in the provided corpus and adding no overhead at inference beyond a single query embedding. [2] also differs in its problem setting, as it begins with real queries to generate synthetic documents for fine-tuning rerankers. In contrast, SQUARE is unsupervised, generates synthetic queries, and adapts embedding models for retrieval. We see these approaches as complementary and SQUARE could potentially generate the initial queries for [2].

Other related works focus on synthetic data generation but for different goals. The primary aim of [20] is to create large, diverse, multilingual datasets for training general, zero-shot embedding models. Conversely, SQUARE focuses on constructing task-specific models by generating synthetic data tailored to a given document set. [1] is much closer to Promptagator [3], focusing on efficient synthetic query generation using small language models. However, it inherits Promptagator's key limitations, namely an inability to effectively handle queries that span multiple documents and a filtering method vulnerable for its reliance on round-trip consistency. Furthermore, its simple generation prompts may limit its ability to create the complex queries necessary for specialized domains.

6 Limitations and Future Works

This paper presents SQUARE, an effective unsupervised framework for customizing retrieval models to specific corpora. By generating and refining synthetic query-answer pairs, SQUARE achieves significant performance improvements compared to baselines.

Although our study utilized state-of-the-art LLMs for the generation of synthetic data, it would be beneficial for future research to investigate the performance of SQUARE when employing smaller, open-source LLMs. This exploration would provide valuable insights into the accessibility and broader applicability of our approach. Furthermore, we intentionally excluded BEIR datasets from our evaluation due to the extensive pre-training of the initial embedding models on these datasets, which could potentially obscure the true impact of SQUARE. Expanding SQUARE's capabilities to include multilingual data expansion is another important avenue for future development.

[2] https://www.anthropic.com/news/claude-3-5-sonnet

References

[1] Tiago Almeida and Sérgio Matos. 2024. Exploring efficient zero-shot synthetic dataset generation for information retrieval. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1214–1231.

[2] Arian Askari, Mohammad Aliannejadi, Chuan Meng, Evangelos Kanoulas, and Suzan Verberne. 2023. Expand, highlight, generate: Rl-driven document generation for passage reranking. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 10087–10099.

[3] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu,

Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.

[4] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

[5] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

[6] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

[7] Krishnaram Kenthapadi, Mehrnoosh Sameki, and Ankur Taly. 2024. Grounding and evaluation for large language models: Practical challenges and lessons learned (survey). In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6523–6533.

[8] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

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

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

[11] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

[12] I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

[13] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

[14] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.

[15] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[16] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

[17] Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S Siegel, Michael Tang, et al. 2024. Bright: A realistic and challenging benchmark for reasoning-intensive retrieval. *arXiv preprint arXiv:2407.12883*.

[18] Weng Lam Tam, Xiao Liu, Kaixuan Ji, Lilong Xue, Xingjian Zhang, Yuxiao Dong, Jiahua Liu, Maodi Hu, and Jie Tang. 2022. Parameter-efficient prompt tuning makes generalized and calibrated neural text retrievers. *arXiv preprint arXiv:2207.07087*.

[19] Voyage. Domain-specific embeddings and retrieval: Legal edition (voyage-law-2). https://blog.voyageai.com/. Accessed: 2025-02-04.

[20] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

[21] Ran Xu, Wenqi Shi, Yue Yu, Yuchen Zhuang, Yanqiao Zhu, May D Wang, Joyce C Ho, Chao Zhang, and Carl Yang. 2024. Bmretriever: Tuning large language models as better biomedical text retrievers. *arXiv preprint arXiv:2404.18443*.

[22] Jinsung Yoon, Yanfei Chen, Sercan Arik, and Tomas Pfister. 2024. Search-adaptor: Embedding customization for information retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12230–12247.

[23] Yizhe Zhang, Siqi Sun, Xiang Gao, Yuwei Fang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2022. Retgen: A joint framework for retrieval and grounded text generation modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11739–11747.

[24] Dawei Zhu, Liang Wang, Nan Yang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. 2024. Longembed: Extending embedding models for long context retrieval. *arXiv preprint arXiv:2404.12096*.

A Generation Prompt

To generate synthetic queries and answers from a document, we use a single, adaptable prompt that works effectively across various datasets.

> Please generate <M> different descriptive queries which includes enough contexts.
> Queries should be answered by the given document but the queries should not refer this document.
> Please avoid straightforward queries and queries should be descriptive.
> Please also generate the answers to these questions that are grounded by the document.
>
> The answer format should be query1 @@@ answer1 /// query2 @@@ answer2 /// ... /// query<M> @@@ answer<M>. Please maintain this format.
>
> Document: <Given Document>

Figure 2: Prompt for synthetic query-answer pair generation. Text in black provides basic instruction; blue text describes output format; and red text is interchangable across different samples.

B Datasets

This section describes the data statistics of the four datasets used in our experiments (see Table 7 and Table 8. It's important to note that only the corpus is used for training; all queries and labels are reserved exclusively for evaluation. Also, this study is limited to English datasets; multilingual datasets are not included.

B.1 BRIGHT

Datasets	BRIGHT-Short			BRIGHT-Long		
	Query no	Corpus no	Label no	Query no	Corpus no	Label no
Biology	103	57539	372	103	524	134
Earth Science	116	121249	585	116	601	187
Economics	103	50220	800	103	516	109
Psychology	101	52835	692	101	512	116
Robotics	101	61961	520	101	508	106
Stackoverflow	117	107081	478	117	1858	129
Sustainable living	108	60792	576	108	554	129
Pony	112	7894	2219	112	577	769

Table 7: Data statistics of BRIGHT-Short and BRIGHT-Long datasets

B.2 LongEmbed & MTEB-Law

LongEmbed				MTEB-Law			
Dataset	Query no	Corpus no	Label no	Dataset	Query no	Corpus no	Label no
Wikim-QA	300	300	300	AILA-casedocs	50	186	195
QMSum	1527	197	1527	AILA-statutes	50	82	217
Summ screen FD	336	336	336	Consumer contracts	396	154	396
Narrative-QA	10449	355	10449	Corporate lobbying	340	319	340
				Legal summary	284	438	439

Table 8: Data statistics of LongEmbed and MTEB-Law datasets.

C Experimental details

C.1 SQUARE

SQUARE generates three synthetic query-answer pairs per document (i.e., M=3 in the prompt). A cutoff threshold of $K = 3$ was used across all experiments; any generated pairs where the relevant document was not within the top-3 retrieved results were discarded. For corpora larger than 10,000 documents, we randomly subsample 10,000 documents for synthetic data generation. This demonstrates SQUARE's efficiency, as it avoids processing the entire corpus, reducing latency and cost. For datasets smaller than 10,000 documents, the entire corpus is used.

C.2 Search-Adaptor tuning

For dense retriever fine-tuning, we primarily use Search-Adaptor [22] due to its effectiveness with limited labeled data. To mitigate overfitting with our unsupervised approach, we use the hyperparameters recommended in the Search-Adaptor paper (specifically, $\alpha = \beta = 0.1$ for all experiments). The synthetic data is split into training (80%) and validation (20%) sets. We then select the Search-Adaptor checkpoint that maximized validation retrieval performance. Critically, both training and validation use only the synthetic data. The held-out test data is used solely for final evaluation.

C.3 LoRA tuning

We experiment with LoRA as an alternative to facilitate efficient fine-tuning of the GTE-Large model [6]. We apply LoRA to all attention layers, set the rank r to 4 and scaling factor a to 8. This configuration resulted in only 589,824 trainable parameters (0.1757% of the model's parameters), reducing the risk of overfitting given that our training dataset is relatively small. Similar to Sec. C.2, the synthetic dataset is split into training and validation sets with an 80/20 ratio and we select the checkpoint that maximized validation retrieval performance. Unlike Search-Adaptor which can be directly applied to the pre-trained embeddings, obtaining LoRA-adapted embeddings requires a full forward pass of the entire model, incurring a much higher computational cost in each validation step, especially when the corpus is large. To reduce the computational time and enable faster evaluation cycles, the validation queries are used to rank a representative subset of documents rather than all the documents. For the loss function, we employ InfoNCE loss [15] with in-batch negatives and fixed the temperature at 0.1. Training is performed using the AdamW optimizer [12] with a learning rate of 5×10^{-5} and a weight decay of 0.01, combined with a linear learning rate scheduler.

C.4 OpenAI Cookbook

We consider the benchmark from the OpenAI cookbook [3], which introduces a method to train a simple projection layer (i.e. a single matrix multiplication) to the pre-trained embeddings for higher accuracy on binary classification tasks. We adopt the approach for our specific use case of semantic similarity-based ranking using GTE-Large embeddings. Specifically, we make the following modifications:

- We compute the retrieval metrics (nDCG@10 or Recall@1) instead of accuracy.
- We replace the originally used MSE loss with the InfoNCE loss [15]. The latter is better fitted for retrieval tasks. Following this, we skip generating negative query-document pairs as our InfoNCE loss inherently utilized in-batch negatives. We use the same temperature (0.1) as Sec. C.3.
- We normalize the projected embeddings with the L2 normalizer. Our experiments showed that this normalization is crucial for the stability of the InfoNCE loss.

A 1024×4096 matrix is used to project the pre-trained embeddings from 1024 dimensions to 4096 dimensions. This follows the recommendation in the cookbook that higher dimensionality yields better results. The matrix is randomly initialized from standard normal distribution. During training, we used the same hyperparameters given in the cookbook: a dropout fraction of 0.2, a batch size of 100 and a

[3] https://cookbook.openai.com/examples/customizing_embeddings

learning rate of 100. Same as other fine-tuning methods, the synthetic dataset is split into training and validation sets with an 80/20 ratio. Matrix selection is based on the checkpoint that achieved the highest validation retrieval performance.

D Additional experiments

D.1 SQUARE with various base LLMs

To further ensure SQUARE works with different LLMs, we test it using Gemini-2.0-Flash-001 to create synthetic query-answer pairs, in addition to our previous experiments with Gemini-1.5-Flash-002 and Claude 3.5 Sonnet v2. We then use Search-Adaptor to fine-tune Google's Gecko-004 embedding models with the SQUARE-generated and filtered synthetic data. As shown in Table 9, different base LLM models yielded high-quality synthetic queries, leading to improved retrieval performance compared to the zero-shot baseline. This validates SQUARE's generalizability across diverse LLMs.

Datasets	Zero	SQUARE	
		Gemini-1.5-Flash	Gemini-2.0-Flash
Biology	0.4035	0.4941	0.4932
Earth-Science	0.4377	0.4931	0.4852
Economics	0.2673	0.3233	0.3080
Psychology	0.3641	0.4208	0.3993
Robotics	0.1746	0.2091	0.1943
Stackoverflow	0.2716	0.2969	0.3055
Sustainable living	0.1687	0.2493	0.2529
Pony	0.0259	0.0218	0.0206
Average	0.2642	0.3136	0.3074

Table 9: Retrieval performance with BRIGHT short datasets in terms of nDCG@10 using two different versions of Gemini (Gemini-1.5-Flash-002 & Gemini-2.0-Flash-001).

D.2 Quantitative evaluation of synthetic query quality

To demonstrate the superior quality of the generated synthetic queries, we propose a strategy that leverages pairwise embedding similarities as a set of quantitative metrics. The objective of this analysis is to show the resemblance between the synthetically generated queries and the original queries. Specifically, we evaluate two key measurements: (i) we calculate the similarity among the original query embeddings themselves, establishing a baseline, (ii) we measure the similarity between the original query embeddings and the synthetic query embeddings. A close alignment between these two sets of similarity scores would provide robust evidence to support our claim that the synthetic queries effectively replicate the intrinsic characteristics of the original query set. Additionally, we also compute the Wasserstein distance between the distribution of original query embeddings and that of the synthetic query embeddings.

Datasets	Embedding similarities		Wasserstein distance
	Among original queries	btw original & synthetic queries	
Wikim-QA	0.3448 ± 0.0999	0.2650 ± 0.0771	1.0668
QMSum	0.4560 ± 0.1398	0.4100 ± 0.1216	0.8656
Summ screen FD	0.4230 ± 0.0824	0.3462 ± 0.0560	1.0255
Narrative-QA	0.3344 ± 0.0654	0.2881 ± 0.0606	1.0743

Table 10: A quantitative analysis of the quality of the generated synthetic queries compared with the original queries.

Table 10 obtained with LongEmbed datasets shows that the embedding similarities among the original queries fall within a range of one standard deviation of the embedding similarities between the synthetically generated queries and the original queries. In addition, Wasserstein distance between the original and synthetic query embeddings is approximately 1.0. We interpret this low distance and the close alignment

of similarity scores as strong indicators of the high quality of the generated synthetic queries and their strong resemblance to the original queries.

D.3 SQUARE with more complex prompt

In the main manuscript, we show the proposed simple prompt in Appendix A achieved consistent performance improvements across the various datasets. We think the proposed SQUARE framework is not sensitive on the prompt. To further demonstrate this point, we provide below more complex prompts along with their corresponding performance metrics.

> You are given a short document about a specific topic. Your ultimate goal is to generate descriptive query and answer pairs based on the document. Please follow the tasks below step by step.
>
> Step 1: analyze what this document is about.
>
> Step 2: based on the document analysis in Step 1, brainstorm a few non-trivial questions with different intents that may be answered after referencing this document and reasoning about it.
>
> Step 3: for the questions in Step 2, provide answers that are grounded in the given document. Explain relevant information and show how it directly relates to the questions. Include supporting data or references directly from the document as needed.
>
> Step 4: combining output above, formulate a synthetic query and answer.
>
> The output format should be: @@@ query @@@ answer. Make sure both the query and answer are descriptive (around 5 sentences respectively) and have sufficient context.
>
> Document: <Given Document>

Figure 3: More complex prompt for synthetic query-answer pair generation. Text in black provides basic instruction; blue text describes output format; and red text is interchangable across different samples.

The results presented in Table 11 and 12 demonstrate that the complex prompt exhibits improved performance on LongEmbed datasets. Conversely, a slight decrease in performance is observed when applied to the BRIGHT-short datasets. It is important to emphasize that both prompts, including the complex one, consistently demonstrate substantial performance improvements when compared to the established zero-shot baselines. This observation strongly suggests that the performance gains observed are not solely attributable to the specific prompt engineering techniques employed, but rather originate from the holistic framework that we have proposed in its entirety.

Datasets	Zero	Current prompt	Complex prompt
Biology	0.4035	**0.4941**	0.4789
Earth-Science	0.4377	**0.4931**	0.4803
Economics	0.2673	**0.3233**	0.3163
Psychology	0.3641	**0.4208**	0.4151
Robotics	0.1746	0.2091	**0.2225**
Stackoverflow	0.2716	**0.2969**	0.2927
Sustainable living	0.1687	**0.2493**	0.2490
Pony	**0.0259**	0.0218	0.0073
Average	0.2642	**0.3136**	0.3078

Table 11: Retrieval performance with BRIGHT short datasets in terms of nDCG@10 using two different prompt.

Datasets	Zero	Current prompt	Complex prompt
Wikim-QA	0.7288	0.7642	**0.8434**
QMSum	0.3830	0.4824	**0.4828**
Summ screen FD	0.9328	0.9542	**0.9747**
Narrative-QA	0.3267	0.4050	**0.4918**
Average	0.5928	0.6515	**0.6982**

Table 12: Retrieval performance with LongEmbed datasets in terms of nDCG@10 using two different prompt.

D.4 Using different LLMs to generate synthetic queries and their corresponding answers

In the main manuscript, Equation (1) illustrates a method where a single LLM call is employed to concurrently generate both the synthetic query and its corresponding answer. In other words, we leverage a single LLM for the generation of both the synthetic query and its associated answer. To explore the potential benefits of using two distinct LLMs, we propose a modification to the process by dividing Equation (1) into a two-stage approach. In the initial stage, a synthetic query is generated based on the provided document. Subsequently, in the second stage, both the original document and the newly generated synthetic query are used to generate the answer. Here, we have implemented this two-stage approach and, as demonstrated below, the observed difference in performance is minimal. We use Gemini-1.5-Flash for synthetic query generation and Claude 3.5 Sonnet v2 for answer generation based on both documents as well as generated synthetic queries.

Datasets	Zero	Query: Gemini-1.5-Flash-002	
		Answer: Gemini-1.5-Flash	Answer: Claude-3.5-Sonnet-v2
Wikim-QA	0.7288	0.7642	0.7608
QMSum	0.3830	0.4824	0.4714
Summ screen FD	0.9328	0.9542	0.9557
Narrative-QA	0.3267	0.4050	0.4086
Average	0.5928	0.6515	0.6491

Table 13: Retrieval performance with LongEmbed datasets in terms of nDCG@10 using the combination of two LLM models for generating synthetic queries and corresponding answers (Gemini-1.5-Flash-002 & Claude-3.5-Sonnet-v2).

Table 13 shows that the two-stage model, in which the synthetic query and answer are generated separately using distinct LLMs, achieves similar performance compared to the one-stage model utilizing Gemini-1.5-Flash-002. Note that the application of unsupervised tuning to both the one-stage and two-stage models yields a substantial improvement in performance relative to the zero-shot baselines.

E Full experimental results

E.1 SQUARE with OpenAI-3-small retriever

BRIGHT-Short			BRIGHT-Long		
Dataset	Zero	SQUARE	Dataset	Zero	SQUARE
Biology	0.3799	**0.4148**	Biology	0.3625	**0.3867**
Earth-Science	0.4777	**0.4807**	Earth-Science	0.3958	**0.4332**
Economics	0.2674	**0.2951**	Economics	0.1942	**0.2767**
Psychology	0.3351	**0.3774**	Psychology	0.2673	**0.3960**
Robotics	0.1525	**0.1873**	Robotics	0.0891	**0.1584**
Stackoverflow	0.2374	**0.2394**	Stackoverflow	0.1496	**0.2137**
Sustainable living	0.2229	**0.2729**	Sustainable living	0.2843	**0.3356**
Pony	**0.0343**	0.0338	Pony	**0.0402**	0.0082
Average	0.2634	**0.2877**	Average	0.2229	**0.2761**

LongEmbed			MTEB-Law		
Dataset	Zero	SQUARE	Dataset	Zero	SQUARE
Wikim-QA	0.6208	**0.6605**	AILA-casedocs	0.3357	**0.3455**
QMSum	0.3312	**0.4780**	AILA-statutes	0.3035	**0.3314**
Summ screen FD	0.8586	**0.9282**	Consumer contracts	0.8004	**0.8112**
Narrative-QA	0.2830	**0.3313**	Corporate lobbying	0.9356	**0.9389**
			Legal summary	0.6868	**0.6941**
Average	0.5234	**0.5995**	Average	0.6124	**0.6242**

Table 14: nDCG@10 retrieval performance (Recall@1 for BRIGHT-Long) across four benchmark datasets using the OpenAI-3-small retriever. We fine-tune the retriever using the Search-Adaptor framework.

E.2 SQUARE with GTE-Large retriever

BRIGHT-Short			BRIGHT-Long		
Dataset	Zero	SQUARE	Dataset	Zero	SQUARE
Biology	0.2317	**0.2686**	Biology	0.4037	**0.4353**
Earth-Science	0.3283	**0.3563**	Earth-Science	0.3815	**0.4749**
Economics	0.2108	**0.2321**	Economics	0.2524	**0.2702**
Psychology	0.2571	**0.2649**	Psychology	0.2297	**0.3584**
Robotics	0.1114	**0.1206**	Robotics	0.1436	**0.1485**
Stackoverflow	0.1518	**0.1608**	Stackoverflow	0.1709	**0.2222**
Sustainable living	0.1651	**0.1997**	Sustainable living	0.2403	**0.2943**
Pony	**0.0374**	0.0323	Pony	0.0059	**0.0095**
Average	0.1867	**0.2044**	Average	0.2285	**0.2767**
LongEmbed			MTEB-Law		
Dataset	Zero	SQUARE	Dataset	Zero	SQUARE
Wikim-QA	0.4917	**0.5623**	AILA-casedocs	**0.3049**	0.2825
QMSum	0.3056	**0.3307**	AILA-statutes	0.2323	**0.2668**
Summ screen FD	0.2241	**0.2951**	Consumer contracts	**0.7424**	0.7110
Narrative-QA	0.6490	**0.7603**	Corporate lobbying	0.9146	**0.9279**
			Legal summary	0.6364	**0.6453**
Average	0.4176	**0.4871**	Average	0.5661	**0.5667**

Table 15: nDCG@10 retrieval performance (Recall@1 for BRIGHT-Long) across four benchmark datasets using the GTE-Large retriever. We fine-tune the retriever using the Search-Adaptor framework.

E.3 Fine-tuning with LoRA using SQUARE datasets

BRIGHT-Short			BRIGHT-Long		
Dataset	Zero	SQUARE	Dataset	Zero	SQUARE
Biology	0.2317	**0.2366**	Biology	**0.4037**	0.3252
Earth-Science	0.3283	**0.3393**	Earth-Science	**0.3815**	0.3772
Economics	**0.2108**	0.2027	Economics	**0.2524**	0.2508
Psychology	**0.2571**	0.2156	Psychology	0.2297	**0.3251**
Robotics	0.1114	**0.1198**	Robotics	**0.1436**	0.1188
Stackoverflow	0.1518	**0.1531**	Stackoverflow	0.1709	**0.2009**
Sustainable living	0.1651	**0.2035**	Sustainable living	**0.2403**	0.2356
Pony	**0.0374**	0.0182	Pony	0.0059	**0.0072**
Average	**0.1867**	0.1861	Average	0.2285	**0.2301**
LongEmbed			MTEB-Law		
Dataset	Zero	SQUARE	Dataset	Zero	SQUARE
Wikim-QA	**0.4917**	0.3178	AILA-casedocs	**0.3049**	0.2844
QMSum	**0.3056**	0.1893	AILA-statutes	0.2323	**0.3341**
Summ screen FD	0.2241	**0.2506**	Consumer contracts	**0.7424**	0.7019
Narrative-QA	**0.6490**	0.5972	Corporate lobbying	0.9146	**0.9225**
			Legal summary	**0.6364**	0.6115
Average	**0.4176**	0.3387	Average	0.5661	**0.5709**

Table 16: nDCG@10 retrieval performance (Recall@1 for BRIGHT-Long) across four benchmark datasets using the GTE-Large retriever. We fine-tune the retriever using LoRA.

E.4 Fine-tuning with OpenAI cookbook using SQUARE datasets

BRIGHT-Short			BRIGHT-Long		
Dataset	Zero	SQUARE	Dataset	Zero	SQUARE
Biology	0.2317	**0.2456**	Biology	0.4037	**0.4523**
Earth-Science	0.3283	**0.3431**	Earth-Science	0.3815	**0.3987**
Economics	**0.2108**	0.1950	Economics	**0.2524**	0.1990
Psychology	**0.2571**	0.2453	Psychology	0.2297	**0.2376**
Robotics	**0.1114**	0.0945	Robotics	**0.1436**	0.0891
Stackoverflow	**0.1518**	0.1408	Stackoverflow	0.1709	**0.1752**
Sustainable living	0.1651	**0.1735**	Sustainable living	0.2403	0.2403
Pony	0.0374	**0.0550**	Pony	**0.0059**	0.0020
Average	**0.1867**	0.1866	Average	**0.2285**	0.2243

LongEmbed			MTEB-Law		
Dataset	Zero	SQUARE	Dataset	Zero	SQUARE
Wikim-QA	**0.4917**	0.4842	AILA-casedocs	0.3049	**0.3164**
QMSum	**0.3056**	0.2683	AILA-statutes	0.2323	**0.2473**
Summ screen FD	0.2241	**0.3233**	Consumer contracts	0.7424	**0.7500**
Narrative-QA	0.6490	**0.6699**	Corporate lobbying	0.9146	**0.9276**
			Legal summary	**0.6364**	0.6240
Average	0.4176	**0.4364**	Average	0.5661	**0.5731**

Table 17: nDCG@10 retrieval performance (Recall@1 for BRIGHT-Long) across four benchmark datasets using the GTE-Large retriever. We fine-tune the retriever based on the OpenAI cookbook.

F Synthetic query examples

Original query	Synthetic query
What is 'protein' in food? I know that proteins perform a lot of functions and as a result there are a lot of different types of proteins in our bodies. When I eat food that has x grams of 'protein', what is this? A homogenous mix of proteins? A heterogenous one? Is it specific set of them that are general enough to be used by my body despite coming from a different organism? How can it be sure that my body can use the kind of protein in the food?	What are the metabolic processes involved in the utilization of dietary protein in animals, and how does this process differ under conditions of nutrient scarcity? How does the body utilize amino acids obtained from food, and what alternative pathways are employed when dietary intake is insufficient to meet energy demands? What are the key enzymes involved in amino acid synthesis, and why are some amino acids considered essential for certain organisms? Finally, what are the consequences of insufficient or poor-quality protein intake in animals?
What is the evolutionary advantage of red-green color blindness? Red-green colorblindness seems to make it harder for a hunter-gatherer to see whether a fruit is ripe and thus worth picking. Is there a reason why selection hasn't completely removed red-green color blindness? Are there circumstances where this trait provides an evolutionary benefit?	What are the different classifications of color blindness based on the severity of the condition and what are the key characteristics of each classification? Furthermore, how common are these different classifications, and are there any notable differences in their symptoms or impact on daily life? Finally, are there any specific genetic factors associated with these different classifications?
We know (think?) that Earth has three Hadley cells per hemisphere, but from observing gas giants such as Jupiter, we see that they have many more cells. According to a link from a comment in this question, Titan might have one cell going from north to south poles. What factors affect the number of cells a planet has, and how? Or to put it another way, given a hypothetical planet with atmosphere, what facts would you need to know in order to guess at how many cells it has?	How does the Hadley circulation contribute to the Earth's overall thermal equilibrium, considering the uneven distribution of solar radiation across different latitudes? Describe the mechanisms involved in this heat redistribution, and explain how this process prevents extreme temperature differences between the equator and the poles. What role do pressure gradients play in driving this circulation? Finally, discuss the significance of latent heat release in this process.
Can GDP ever be negative? Imagine that in an economy we only produced a toy and that cost us 10 dollars. We don't sell this toy in year 1. Then this goes under Investment as +10 and GDP is +10. But say the only thing we do in year 2 is sell this toy for a loss at 8 dollars. Then investment is now -10 but consumption is +8 so GDP in year 2 is -2. Is this a valid example of negative GDP?	How does the change in private inventories impact the Gross Domestic Product (GDP), considering different scenarios of inventory accumulation and decumulation over consecutive periods? Explain the three possible ways an increase in GDP can occur due to changes in the level of private inventories, providing examples of how inventory levels might shift to cause these increases. What are the key accounting principles involved in measuring these changes?
Why can't MEG distinguish EPSPs and IPSPs? I see this fact stated in many places that discuss the basis of the MEG signal but it's never been made completely clear why MEG can't distinguish IPSPs and EPSPs. Please could someone elaborate?	What are the challenges in surgical treatment of epilepsy, and how can non-invasive brain imaging techniques contribute to improving surgical outcomes and minimizing risks to patients undergoing such procedures? The need to precisely identify the seizure origin is crucial to avoid damaging healthy brain tissue during surgery. What role does advanced neuroimaging play in this process, and how does it compare to traditional invasive methods? How might this technology potentially reduce the need for invasive procedures?
Comparing navigation algorithms I am new to Nav2 stack and currently there are a bunch of navigation plugins to choose from (https://navigation.ros.org/plugins/index.html). I am interested in how to choose available algorithms from those in "Planners" and "Controllers" sections. For example, how to know pros and cons of NavFn planner compared with SmacPlanner. How to know pros and cons of DWB controller against TEB controller. Is there any site clarify something like this?	Imagine a scenario involving a high-speed robot tasked with delivering packages. What sophisticated planning techniques are employed to ensure the robot avoids obstacles, maintains path smoothness, and prevents accidents like tipping over or dropping its load, especially considering the robot's speed and the potential for centripetal forces to destabilize it? Describe the methods used to optimize the robot's path planning and the benefits of these methods. Finally, discuss how the planning system addresses the challenges posed by high-speed movement and the need for safe navigation.

Table 18: The table shows examples of a relevant query and a synthetic query generated by SQUARE for the same document. These examples allow for a qualitative assessment of SQUARE's generated queries, specifically examining their similarity to the original query.