Benchmarking the Myopic Trap: Positional Bias in Information Retrieval

Anonymous ACL submission

Abstract

This study investigates a specific form of positional bias—termed the *Myopic Trap*—where retrieval models disproportionately attend to the early parts of documents while overlooking relevant information that appears later. To systematically quantify this phenomenon, we propose a semantics-preserving evaluation framework that repurposes the existing NLP datasets into position-aware retrieval benchmarks. By evaluating the SOTA models of full retrieval pipeline—including BM25, embedding models, ColBERT-style late-interaction models, and reranker models—we offer a broader empirical perspective on positional bias than prior work. Experimental results show that embedding models and ColBERT-style models exhibit significant performance degradation when query-related content is shifted toward later positions, indicating a pronounced head bias. Notably, under the same training configuration, ColBERT-style approach show greater potential for mitigating positional bias compared to the traditional single-vector approach. In contrast, BM25 and reranker models remain largely unaffected by such perturbations, underscoring their robustness to positional bias.[1]

1 Introduction

Information Retrieval (IR) systems serve as the foundation for a wide range of applications, including web search engines (Croft et al., 2010; Huang et al., 2020), question answering (Tellex et al., 2003), and Retrieval-Augmented Generation (RAG) (Lewis et al., 2020). A core challenge in IR systems lies in accurately evaluating the semantic relevance between user queries and candidate documents. However, biases in retrieval models could inadvertently distort this relevance estimation, impacting the accuracy of IR systems (Lipani, 2019).

[1]To facilitate further research, we release all code and datasets at: https://github.com/xxx.

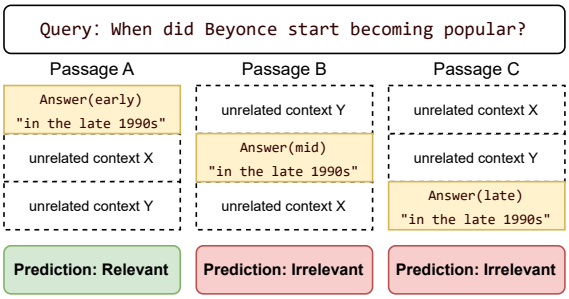

Figure 1: Illustration of the Myopic Trap, where retrieval models exhibit a strong positional bias by undervaluing passages with relevant content in later sections.

This work investigates a specific form of positional bias in IR, which we term the *Myopic Trap*—the tendency of retrieval models to disproportionately favor information near the beginning of documents, while overlooking relevant content that appears later, thus underestimating the overall document relevance (Figure 1). Prior studies have identified early signs of such behavior in a limited number of embedding models (Coelho et al., 2024; Fayyaz et al., 2025). However, it remains unclear whether increasingly powerful open-source embedding models exhibit similar tendencies. At the same time, a broader evaluation of positional bias across the full IR pipeline is crucial for understanding and improving end-to-end retrieval systems.

Current evaluation paradigms for positional bias often depend on synthetic modifications to documents, such as the insertion of relevant spans at predetermined positions (Coelho et al., 2024; Fayyaz et al., 2025). While such controlled settings facilitate analysis, they risk introducing artifacts that compromise the realism of evaluations. To address this issue, we propose a novel evaluation framework for assessing positional bias that preserves the original content of documents, by repurposing two NLP datasets: SQuAD v2 (Rajpurkar et al., 2018) and FineWeb-edu (Penedo et al., 2024).

From SQuAD v2, we construct a position-aware retrieval task by grouping questions based on the location of their corresponding answer spans within documents. From FineWeb-edu, we generate fine-grained, position-sensitive questions using Large Language Models (LLMs) (Zhao et al., 2025; Long et al., 2024), with each question targeting a specific content segment—*beginning*, *middle*, and *end*.

We conduct experiments on a range of state-of-the-art (SOTA) retrieval models—including BM25, embedding models, ColBERT-style late interaction models, and reranker models—to ensure that our analysis of positional bias is grounded in the best-performing systems available today. Experimental results show that embedding models and ColBERT-style models exhibit significant performance degradation when relevant content appears later in the document. Notably, under the same training configuration, ColBERT-style approach show greater potential for mitigating positional bias compared to the traditional single-vector approach. In contrast, BM25 and reranker models demonstrate greater robustness to positional bias, benefiting from either exact token matching or deep cross-attention mechanisms that better localize query-relevant information.

To the best of our knowledge, this is the first study to systematically investigate positional bias across the full IR pipeline, particularly with state-of-the-art retrieval models, thereby shedding light on both its associated risks and potential mitigation strategies.

2 Related Work

Positional bias—where models disproportionately focus on specific segments of a document—has garnered increasing attention in the IR community. Coelho et al. (2024) first report that embedding models exhibit a pronounced primacy bias, encoding early document content more effectively than later parts. They show that this bias originates during contrastive pretraining and is further amplified through fine-tuning on datasets like MS MARCO (Nguyen et al., 2016), using models such as T5 (Raffel et al., 2020) and RepLLaMA (Ma et al., 2024). They also highlight a structural characteristic of MS MARCO: an uneven distribution of information density, with relevant content disproportionately concentrated at the beginning of documents. Building on this line of work, Fayyaz et al. (2025) repurpose a relation extraction dataset to study multiple forms of bias in embedding models. In addition to primacy bias, they identify tendencies such as a preference for shorter documents, repetition of matching entities, and reliance on literal string matches. Importantly, they demonstrate how these biases can be exploited to manipulate RAG systems, ultimately prompting LLMs to produce harmful or misleading content.

3 Experiments

3.1 Position-Aware Retrieval Tasks

To investigate the *Myopic Trap* phenomenon, we construct two position-aware retrieval tasks that quantify positional bias across various retrieval models. The corresponding dataset statistics, construction processes, and prompt engineering details are presented in Appendix B and C.

3.1.1 Repurposing Existing QA Datasets

We repurpose the Stanford Question Answering Dataset v2 (SQuAD v2), leveraging its character-level answer span annotations to enable fine-grained positional analysis. After removing unanswerable questions—originally designed to probe abstention behaviors—we obtain 92,749 examples, each represented as a (*question*, *passage*, *answer_start_position*) triple. We denote this dataset as **SQuAD-PosQ-Full**. To examine positional bias, we bucket the questions into six groups based on the character-level start index of their answers: [0–100], [100–200], [200–300], [300–400], [400–500], and [500–3120][2], with all bins inclusive. Retrieval is framed as a passage ranking task over the set of all unique passages. A consistent decline in performance for questions whose answers appear later in the document would indicate the Myopic Trap. For scalability, we also construct a smaller subset, **SQuAD-PosQ-Tiny**, containing 10,000 triples sampled randomly, while keeping the retrieval corpus unchanged.

3.1.2 Synthetic Position-Aware Questions

While SQuAD-PosQ serves as a useful benchmark, it has two key limitations: (1) its passages are relatively short (averaging 117 words), and (2) it is likely included in the training data of many retrieval models, raising concerns about evaluation leakage (Chen et al., 2024; Lee et al., 2025). To address these issues, we construct a synthetic dataset using

[2]The maximum observed index is 3120.

Table 1: NDCG@10 scores of retrieval models on SQuAD-PosQ and FineWeb-PosQ. Models with significant positional bias are underscored; some model names are abbreviated for display clarity.*

Retrieval Models	SQuAD-PosQ						FineWeb-PosQ		
	0+	100+	200+	300+	400+	500+	begin	middle	end
BM25	76.62	79.37	80.61	81.06	81.43	79.49	89.56	89.63	88.80
Embedding Models									
bge-m3-dense	84.47	83.03	81.47	79.95	77.98	74.61	88.64	84.75	80.35
stella_en_400M_v5	85.78	83.62	82.24	80.34	78.96	75.69	88.19	83.93	78.96
voyage-3-large	89.93	89.32	89.17	88.70	88.09	86.73	92.65	90.63	87.96
text-embedding-3-large	85.19	82.45	80.32	77.84	75.27	71.10	86.09	83.84	82.09
gte-Qwen2-7B-instruct	85.13	83.85	83.33	81.71	80.13	77.75	87.45	84.92	81.79
NV-embed-v2	93.04	93.55	93.48	93.02	92.48	90.72	87.35	88.39	88.10
ColBERT-style Models									
colbertv2.0	91.85	90.27	91.74	89.64	86.71	84.57	88.73	77.78	64.25
bge-m3-colbert	89.88	88.09	88.84	87.68	86.72	86.36	92.08	90.23	86.66
ReRankers									
bge-reranker-v2-m3	93.53	93.56	94.69	94.50	94.42	94.52	95.18	95.21	94.66
gte-reranker-base*	90.70	91.10	92.59	91.84	91.57	92.03	95.43	95.74	95.41
bge-reranker-gemma*	94.31	94.01	94.73	94.80	94.55	94.55	95.56	95.73	95.46

passages from the FineWeb-edu, a large-scale, high-quality educational web text corpus. We sample 13,902 passages ranging from 500 to 1,024 words, and use `gpt-4o-mini` (OpenAI, 2024a) to generate global summaries and position-aware question–answer pairs grounded in localized chunks of each passage. We filtered out responses that did not match the expected output format and manually reviewed 100 randomly selected ones, finding no significant anomalies. Each passage is divided into three equal-length segments—*beginning*, *middle*, and *end*—and each question is tagged according to the location of its supporting chunk. If a chunk spans two segments, we assign both tags to reflect ambiguity. The resulting dataset, **FineWeb-PosQ-Full**, enables robust evaluations of retrieval models in long-form, position-sensitive contexts. We also create a smaller version, **FineWeb-PosQ-Tiny**, by sampling approximately 3,300 questions per segment category, resulting in 6,620 unique questions after deduplication.

3.2 Experimental Results

To assess susceptibility to the Myopic Trap, we conduct a comprehensive evaluation across the full IR pipeline, covering retrieval models from four distinct categories.

- **Probabilistic Models**: BM25 (Robertson et al., 1994)
- **Embedding Models**: bge-m3-dense[3] (Chen et al., 2024), stella_en_400M_v5 (Zhang et al., 2025), text-embedding-3-large (OpenAI, 2024b), voyage-3-large (VoyageAI, 2025), gte-Qwen2-7B-instruct (Li et al., 2023b), NV-embed-v2 (Lee et al., 2025)
- **ColBERT-style Models**: colbertv2.0 (Santhanam et al., 2022), bge-m3-colbert[4] (Chen et al., 2024)
- **ReRankers**: bge-reranker-v2-m3 (Chen et al., 2024), gte-multilingual-reranker-base (Zhang et al., 2024), bge-reranker-v2-gemma (Li et al., 2023a)

We adopt NDCG@10 as our primary evaluation metric, which captures both retrieval accuracy and ranking quality within the top-10 retrieved results. To control computational costs, BM25 and embedding models are evaluated on the full datasets, whereas the computation-intensive ColBERT-style and reranker models are assessed on the tiny subsets. Experimental results are presented in Table 1, followed by an in-depth analysis.

[3] `bge-m3-dense` denotes the dense retrieval mode of the `bge-m3` model, where a single vector is generated per query or document.

[4] `bge-m3-colbert` refers to the late interaction mode of the `bge-m3` model, where multiple token-level embeddings are generated for each input to enable ColBERT-style retrieval.

3.2.1 BM25: Natural Immunity

BM25, a classical sparse retrieval method grounded in term-matching statistics, exhibits strong robustness to positional bias across both datasets. Its NDCG@10 scores remain stable across different answer positions within passages. This behavior is expected, as BM25 relies solely on keyword matching and does not consider term positions within documents. While such position-agnostic behavior may be a limitation in retrieval tasks requiring an understanding of discourse structure or semantic coherence, it proves advantageous in retrieval scenarios affected by positional bias.

3.2.2 Embedding Models: Vulnerability

Experimental results on a broad range of modern embedding models confirm and extend the findings of Coelho et al. (2024); Fayyaz et al. (2025). We observe that the Myopic Trap is a widespread issue affecting both open-source and commercial embedding models, including large-scale architectures like `gte-Qwen2-7B-instruct`. Across both datasets, retrieval performance consistently deteriorates as relevant information appears later in the passage. Coelho et al. (2024) attribute this bias to contrastive pretraining, which is further amplified during contrastive fine-tuning. Given that contrastive learning remains the predominant strategy for training supervised embedding models, our findings underscore the urgent need to reassess how such training pipelines contribute to positional bias. A notable exception in our evaluations is `NV-embed-v2` and `voyage-3-large`, which demonstrate relatively strong robustness to positional variance. We suspect that the latent attention layer of `NV-embed-v2`, designed to support more expressive sequence pooling, may help preserve global contextual information and thereby partially mitigate the effects of the Myopic Trap.

3.2.3 ColBERT-style Models: Bias Persistence

ColBERT (Khattab and Zaharia, 2020) and similar late-interaction models strike a balance between retrieval efficiency and effectiveness by independently encoding queries and documents into multi-vector representations, performing token-level interactions only during final scoring. Despite their advanced interaction design, our results show that ColBERT-style models still suffer from the Myopic Trap across both datasets. This suggests that although late-stage token interactions improve relevance estimation, they are insufficient to fully mitigate positional bias likely introduced during earlier encoding stages. A particularly intriguing case lies in the contrast between `bge-m3-dense` and `bge-m3-colbert`. Although their vector representations are derived from the same base model, `bge-m3-colbert` demonstrates significantly greater robustness to positional bias. This finding suggests that, under the same training configuration, the ColBERT-style training approach is more effective at mitigating positional bias than the traditional single-vector retrieval approach.

3.2.4 Reranker Models: Effective Mitigation

Reranker models, which apply full query–passage interaction via deep cross-attention, are typically used only on a small set of first-stage candidates due to their computational cost. Our experiments show that such models consistently mitigate the Myopic Trap across both datasets and model scales. The cross-attention mechanism enables precise identification of relevant content regardless of its position in the passage, effectively neutralizing positional bias when in earlier retrieval stages. This has important implications for IR system design: while embedding-based and ColBERT-style retrievers may introduce positional biases—especially when relevant content appears later—a reranking stage can substantially correct for these issues. In position-sensitive applications such as RAG, incorporating a reranker provides a strong safeguard against relevance degradation caused by positional effects and is thus essential for building a fair and reliable retrieval system.

4 Conclusion

This study investigates the *Myopic Trap* bias across the full IR pipeline, including BM25, embedding models, ColBERT-style late-interaction models, and reranker models. We heuristically construct semantics-preserving, position-aware retrieval benchmarks by repurposing existing NLP datasets, enabling a systematic evaluation of this bias. Using these benchmarks, we present the first comprehensive, pipeline-wide analysis of the Myopic Trap, providing an empirical perspective on how such bias emerges across the retrieval stack. Our findings show that while the Myopic Trap originates in embedding-based retrievers, it can be substantially mitigated by downstream interaction-based rerankers.

Limitation

This work has several limitations that open avenues for future research. First, our study focuses exclusively on English-language text retrieval. Positional bias in multilingual and cross-lingual retrieval settings remains unexplored and warrants further investigation. Second, while we use LLMs to generate synthetic question–answer pairs grounded in passages and apply manual quality control, some degree of noise may still persist. In future work, we aim to improve data quality through multi-agent collaboration and more robust verification pipelines. Third, our analysis does not yet offer a theoretical explanation for why embedding models exhibit uneven information distribution in their vector representations. We plan to explore embedding theory more deeply in future work, with the goal of informing more robust and unbiased text representation learning.

References

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

João Coelho, Bruno Martins, Joao Magalhaes, Jamie Callan, and Chenyan Xiong. 2024. Dwell in the beginning: How language models embed long documents for dense retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 370–377, Bangkok, Thailand. Association for Computational Linguistics.

W Bruce Croft, Donald Metzler, and Trevor Strohman. 2010. *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Reading.

Mohsen Fayyaz, Ali Modarressi, Hinrich Schütze, and Nanyun Peng. 2025. Collapse of dense retrievers: Short, early, and literal biases outranking factual evidence. *CoRR*, arXiv:2503.05037.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 2553–2561, New York, NY, USA. Association for Computing Machinery.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA. Association for Computing Machinery.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. NV-embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023a. Making large language models a better foundation for dense retrieval. *Preprint*, arXiv:2312.15503.

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

Aldo Lipani. 2019. On biases in information retrieval models and evaluation. *SIGIR Forum*, 52(2):172–173.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 11065–11082. Association for Computational Linguistics.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2421–2425, New York, NY, USA. Association for Computing Machinery.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

OpenAI. 2024a. Gpt-4o mini: advancing cost-efficient intelligence.

OpenAI. 2024b. New embedding models and api updates.

Guilherme Penedo, Hynek Kydlícek, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin A. Raffel, Leandro von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST).

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 28 - August 1, 2003, Toronto, Canada*, pages 41–47. ACM.

VoyageAI. 2025. voyage-3-large: the new state-of-the-art general-purpose embedding model.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. 2025. Jasper and stella: distillation of sota embedding models. *Preprint*, arXiv:2412.19048.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. 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, Miami, Florida, US. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2025. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Distribution Analysis on SQuAD v2

Figure 2 shows the distribution of answer start positions in SQuAD v2, which follows a pronounced long-tail pattern: answers tend to appear near the beginning of passages, though a substantial portion also occurs in later positions. This natural distribution makes SQuAD v2 particularly well-suited for studying positional effects in retrieval models.

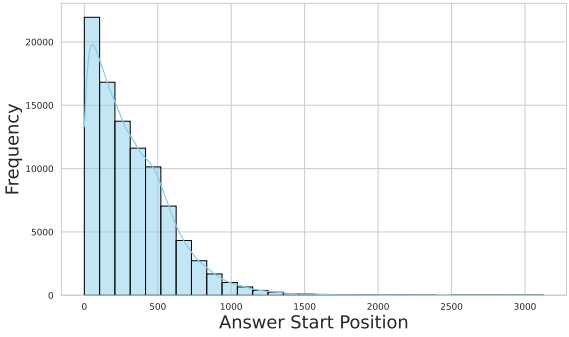

Figure 2: Distribution of answer start positions in SQuAD v2.

B Dataset Cards

We detail the construction of the SQuAD-PosQ and FineWeb-PosQ datasets below. Detailed statistics for the two datasets are provided in Table 2.

B.1 SQuAD-PosQ

Stanford Question Answering Dataset v2 (SQuAD v2) is a reading comprehension dataset where each instance comprises a question, a context passage, and, for answerable questions, the corresponding answer span. Crucially, SQuAD v2 provides the character-level start index of each answer, enabling

Table 2: Statistics of SQuAD-PosQ and FineWeb-PosQ Datasets.

	SQuAD-PosQ-Full	*-Tiny	FineWeb-PosQ-Full	*-Tiny
# Query	92,749	10,000	265,865	6,620
Mean Query Length	10.09	10.08	14.06	13.99
Std Query Length	3.56	3.56	4.30	4.23
# Passage	20,233	–	13,902	–
Min Passage Length	20	–	500	–
Mean Passage Length	117.19	–	707.38	–
Max Passage Length	653	–	1,023	–
Std Passage Length	50.22	–	146.47	–
Position Group				
0+: [0–100]	21,220	2,252	begin: 103,844	begin: 3,300
100+: [100–200]	16,527	1,813		
200+: [200–300]	13,667	1,444	middle: 199,742	middle: 5,098
300+: [300–400]	11,514	1,210		
400+: [400–500]	10,089	1,108	end: 160,832	end: 3,300
500+: [500–3120]	20,384	2,237		

Table 3: Examples from the FineWeb-PosQ dataset with corresponding position tags.

No.	Question	Position Tag(s)
1	Where was Lynne Frederick born and raised, and who raised her?	[beginning, middle]
2	What is unique about Angkor Wat's history compared to other Angkor temples, considering its post-16th century status?	[before, middle]
3	What was Doris Speed known for doing on set to lighten the mood?	[after]
4	Why might the competition to determine the county with the longest coastline be driven more by tourism than by definitive geographical data?	[middle, after]
5	What should be considered important in the delivery of a persuasive speech?	[after]

fine-grained positional analysis by stratifying questions based on the location of the answer within the passage. We begin by merging the official training and validation sets, excluding all unanswerable (adversarially constructed) instances, as our focus is on contexts where answers are present at varying positions. The resulting dataset contains 92,749 answerable examples, each represented as a (query, passage, answer_start_position) triple. We refer to this as **SQuAD-PosQ-Full**. To quantify positional bias, we stratify SQuAD-PosQ-Full into six bins by character-level answer start index: [0–100], [100–200], [200–300], [300–400], [400–500], and [500–3120], where 3120 is the maximum observed index and bins are inclusive. To facilitate efficient evaluation, we construct **SQuAD-PosQ-Tiny**, randomly sampled 10,000 triples from SQuAD-PosQ-Full, with the retrieval corpus kept fixed.

B.2 FineWeb-PosQ

FineWeb-edu is a large-scale, high-quality educational web text corpus. We begin by selecting 13,902 passages from FineWeb-edu, filtering for those with word counts between 500 and 1024 to ensure sufficient length. Each passage is summarized by `gpt-4o-mini` to provide global context for question generation, and then segmented into 256-word chunks using the RecursiveCharacterTextSplitter[5] to support location-specific question creation. For each chunk, the LLM generates a (question, answer, question_type) triplet, using both the chunk and its corresponding global summary as input. Our initial approach involved generating questions alone; however, manual inspection revealed that approximately 40% were unanswer-

[5] https://python.langchain.com/docs/how_to/recursive_text_splitter/

able or misaligned. To improve answerability and relevance, we revised the prompt to require simultaneous generation of both question and answer, ensuring extractability from the given chunk. Despite directly prompting the LLM to generate "complex" questions, we observed a tendency toward simpler forms, such as basic entity recognition. To encourage greater complexity, we introduced a `question_type` field (either *simple* or *complicated*) in the prompt. While this field helped guide generation, we do not use it for filtering or analysis—all valid questions are retained regardless of type. To ensure data quality, we filtered out responses that did not match the expected output format and manually reviewed 100 randomly selected generation traces, finding no significant anomalies. To encode positional information, each passage is divided into three equally sized segments: *beginning*, *middle*, and *end*. Each question is tagged according to the segment containing its source chunk (Algorithm 1). If a chunk spans multiple segments, the corresponding question is tagged with both. The resulting dataset, **FineWeb-PosQ-Full**, supports position-sensitive retrieval tasks over longer texts. Example questions are shown in Table 3. For efficient evaluation, we construct **FineWeb-PosQ-Tiny** by randomly sampling 3,300 questions from each positional category. After deduplication, the final subset contains 6,620 unique questions.

Algorithm 1 POSITION TAGGING

Require: Total length z, chunk start index m, end index n

Ensure: Return tag(s): `beginning`, `middle`, `end`

$third \leftarrow \lfloor z/3 \rfloor$
if $n < third$ **then**
  return { `before` }
else if $m \geq third$ **and** $n < 2 \cdot third$ **then**
  return { `middle` }
else if $m \geq 2 \cdot third$ **then**
  return { `after` }
else if $n < 2 \cdot third$ **then**
  return { `before`, `middle` }
else
  return { `middle`, `after` }
end if

B.3 Validity of the Sampled Subset

To empirically verify the validity of the sampled dataset (i.e., SQuAD-PosQ-Tiny and FineWeb-PosQ-Tiny), we conduct preliminary experiments using `bge-m3-dense` on both the full and tiny versions of each dataset. As shown in Figure 3, `bge-m3-dense` demonstrates highly consistent performance between the full and sampled datasets, particularly on FineWeb-PosQ, under the NDCG@10 metric. These results confirm the feasibility of using the sampled subset to accelerate evaluation for computationally intensive models. The results also reveal the pronounced Myopic Trap bias in `bge-m3-dense`, indicating a tendency to overly prioritize the beginning context during retrieval.

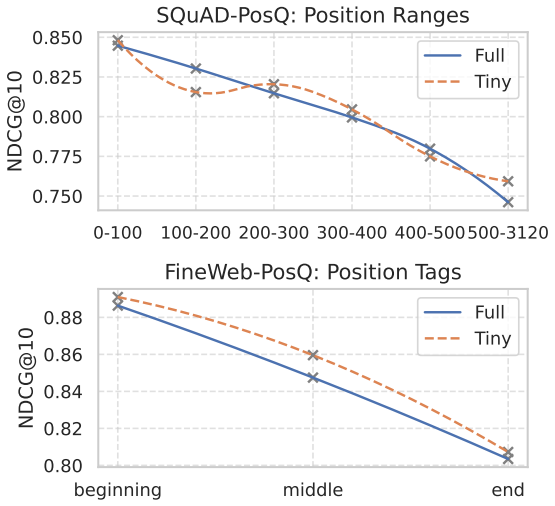

Figure 3: NDCG@10 scores of `bge-m3-dense` on Full vs. Tiny Datasets.

B.4 Representation Behavior

Following the approach of Coelho et al. (2024), we compute the cosine similarity between the *full-text* embedding and the embeddings of the *beginning*, *middle*, and *end* segments to examine how embedding models represent different parts of the text. We selected a random subset of 10,000 passages from the SQuAD v2 dataset (with lengths ranging from 100 to 512 words, average 146 words) and 10,000 passages from the FineWeb-Edu dataset (with lengths ranging from 200 to 500 words, average 339 words). As shown in Table 4, we observe that the similarity between the beginning segment and the full text is consistently the highest across most models. This suggests that although these models are designed to encode the entire input, they tend to overemphasize its initial portion. In contrast, similarity scores for the middle and end segments show a noticeable decline. For instance,

Table 4: Cosine similarity between full-text embeddings and segment-level embeddings (beginning, middle, end) across models and datasets. Higher values indicate stronger alignment between the segment and the full-text representation.

Dataset	Embedding Model	Full & Begin	Full & Middle	Full & End
SQuAD v2	bge-m3-dense	0.8777	0.7957	0.7727
	stella_en_400M_v5	0.8851	0.8188	0.7930
	text-embedding-3-large	0.8695	0.7451	0.7251
	voyage-3-large	0.8695	0.8446	0.8335
	gte-Qwen2-7B-instruct	0.8440	0.7831	0.7456
	NV-Embed-v2	0.7760	0.7058	0.6854
FineWeb-Edu	bge-m3-dense	0.9201	0.8101	0.7835
	stella_en_400M_v5	0.9255	0.8514	0.8280
	text-embedding-3-large	0.8977	0.7444	0.7805
	voyage-3-large	0.9278	0.8837	0.8712
	gte-Qwen2-7B-instruct	0.8683	0.7775	0.7821
	NV-Embed-v2	0.8430	0.7402	0.7651

in `text-embedding-3-large`, the similarity drops from 0.8695 (full & beginning) to 0.7451 (full & middle), and further to 0.7251 (full & end). This tendency is consistent across many models, reinforcing the observation that embedding models exhibit a strong positional bias—favoring the beginning of the input while underrepresenting its later parts.

C Prompts

C.1 Prompt for Summarization

<Task>
Given a document, please paraphrase it concisely.
</Task>

<Requirements>
- The paraphrase should be concise but not missing any key information.
- Please decide the number of words for the paraphrase based on the length and content of the document, but do not exceed 400 words.
- You MUST only output the paraphrase, and do not output anything else.
</Requirements>

<Document> `{TEXT}` </Document>

C.2 Prompt for Question Generation

<Task>
Given a summary and a chunk of document, please brainstorm some FAQs for this chunk.
</Task>

<Requirements>
- The generated questions should be high-frequency and commonly asked by people.
- Two types of questions should be generated: simple (e.g., factual questions) and complicated (questions that require reasoning and deep thinking to answer).
- The majority of the questions you generate should be complicated.
- The answers to the questions must be based on the chunk and should not be fabricated.
- You MUST only output the FAQs, and do not output anything else.
Note: The FAQ you generate must be based on this chunk rather than the summary!!! The summary is only used to assist you in understanding the chunk.
</Requirements>

<summary> `{SUMMARY}` </summary>

<chunk> `{CHUNK}` </chunk>

<Output Format>
Your output should be a JSON List:

```
[
  {
    "question": "Genrated question",
    "answer": "The answer of question",
    "type": "simple or complicated"
  },
  ...
]
```

</Output Format>