An Empirical Study of Position Bias in Modern Information Retrieval

Ziyang Zeng[1,2] **Dun Zhang**[2,3*] **Jiacheng Li**[2]
Panxiang Zou[4] **Yudong Zhou**[3] **Yuqing Yang**[1†]
[1]Beijing University of Posts and Telecommunications
[2]NovaSearch Team [3]Prior Shape [4]RichInfo
ziyang1060@bupt.edu.cn, {dunnzhang0,jcli.nlp}@gmail.com
zoupanxiang@richinfo.cn, zhouyudong@priorshape.com
yangyuqing@bupt.edu.cn

Abstract

This study investigates the *position bias* in information retrieval, where models tend to overemphasize content at the beginning of passages while neglecting semantically relevant information that appears later. To analyze the extent and impact of position bias, we introduce a new evaluation framework consisting of two position-aware retrieval benchmarks (SQuAD-PosQ, FineWeb-PosQ) and an intuitive diagnostic metric, the Position Sensitivity Index (PSI), for quantifying position bias from a worst-case perspective. We conduct a comprehensive evaluation across the full retrieval pipeline, including BM25, dense embedding models, ColBERT-style late-interaction models, and full-interaction reranker models. Our experiments show that when relevant information appears later in the passage, dense embedding models and ColBERT-style models suffer significant performance degradation (an average drop of 15.6%). In contrast, BM25 and reranker models demonstrate greater robustness to such positional variation. These findings provide practical insights into model sensitivity to the position of relevant information and offer guidance for building more position-robust retrieval systems. Code and data are publicly available at: https://github.com/NovaSearch-Team/position-bias-in-IR.

1 Introduction

Information Retrieval (IR) underpins a broad range of applications, such as web search (Croft et al., 2010), question answering (Tellex et al., 2003), and Retrieval-Augmented Generation (RAG) (Lewis et al., 2020). A central challenge for IR systems is to accurately assess the semantic relevance between user queries and candidate passages. Recent advances in neural IR models, particularly those leveraging pre-trained language models such as BERT (Devlin et al., 2019), have significantly improved retrieval performance (Yates et al., 2021; Gao et al., 2021). However, prior work (Hofstätter et al., 2021; Jiang et al., 2021; MacAvaney et al., 2022) has identified that such models exhibit a *position bias*: a tendency to overemphasize content at the beginning of passages, while overlooking semantically relevant information appearing later. Figure 1 illustrates how such position bias can lead retrieval models to systematically underestimate the relevance of passages, especially when key information appears later, which may harm downstream performance (e.g., in RAG (Fayyaz et al., 2025)) or expose vulnerabilities to adversarial attacks (Wang et al., 2022).

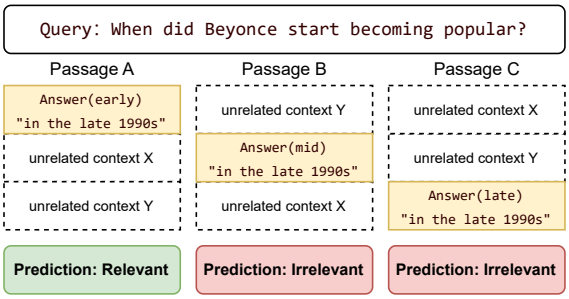

Figure 1: Illustration of position bias in IR: models often focus disproportionately on the beginning of passages, overlooking relevant content that appears later.

While modern neural IR models have seen significant advances in architecture and training techniques (Ma et al., 2024; Lee et al., 2025), recent studies suggest that position bias remains present in a few specific embedding models (Coelho et al., 2024; Fayyaz et al., 2025). This motivates a broader research question: *How prevalent is position bias among today's state-of-the-art IR models, and how does this bias manifest across different IR architectures?* Answering this question requires a systematic investigation across diverse classes of retrieval models—an area that remains largely un-

[*] Project leader.
[†] Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 5069–5081
November 4-9, 2025

derexplored. Meanwhile, the findings from recent studies on position bias are often based on synthetically manipulated passages—e.g., by inserting relevant spans at predetermined positions (Coelho et al., 2024; Fayyaz et al., 2025). While these controlled setups are diagnostically useful, they risk introducing semantic discontinuities and may not reflect realistic retrieval conditions.

To this end, we introduce two new position-aware English retrieval benchmarks—SQuAD-PosQ and FineWeb-PosQ—designed to evaluate model performance in retrieval scenarios where relevant content appears at varying positions within passages. Derived from SQuAD v2 (Rajpurkar et al., 2018) and FineWeb-edu (Penedo et al., 2024), respectively, these benchmarks differ in passage length and construction methodology, offering complementary perspectives for analysis. We preserve the original passage structure and construct position-sensitive questions either from existing annotated QA pairs or by prompting large language models (LLMs) (Zhao et al., 2025) to target specific passage regions. A two-stage filtering pipeline is applied to validate positional relevance and minimize false negatives (Chen et al., 2025) during question generation. To accompany these benchmarks with a quantifiable diagnostic, we propose a simple and intuitive metric, the Position Sensitivity Index (PSI), which provides a worst-case perspective by explicitly quantifying the maximum relative degradation across positions.

We perform a comprehensive evaluation across diverse IR models to analyze the extent and impact of position bias, including sparse retrievers (e.g., BM25), dense embedding-based retrievers, ColBERT-style late-interaction models, and full-interaction reranker models. Our results reveal that dense embedding and ColBERT-style models exhibit an average 15.6% performance drop when relevant content is located later in the passage, revealing a consistent bias toward early-passage content. In contrast, BM25 and reranker models remain largely robust to positional shifts. These differences highlight architectural sensitivities to where relevant information appears in a passage.

2 Related Work

Prior work has extensively documented the existence of position bias in neural IR systems. Hofstätter et al. (2021) observe that in MS MARCO (Nguyen et al., 2016), a widely used collection in the IR community, answer spans are disproportionately concentrated in the earlier portions of passages. BERT-based neural IR models fine-tuned on MS MARCO tend to inherit and reinforce this position bias (Jiang et al., 2021; MacAvaney et al., 2022), potentially leading to overestimated performance due to shared distributional artifacts between training and evaluation sets (Rau et al., 2024). Coelho et al. (2024) further dissect the training pipeline of a T5-based dense retriever on MS MARCO, showing that position bias primarily originates during contrastive pre-training and is amplified in contrastive fine-tuning. Fayyaz et al. (2025) extend this analysis by uncovering multiple forms of bias—including position bias—in dense embedding models, and linking them to generation failures in RAG pipelines. Beyond IR, similar position-sensitive behaviors have been observed in LLMs, particularly in how attention is distributed across long contexts (Liu et al., 2024; Zhang et al., 2025c; Wu et al., 2025), suggesting that position bias may be a more general limitation of modern transformer architectures.

3 Position-Aware Retrieval Benchmarks

3.1 Repurposing Existing QA Pairs

We repurpose the Stanford Question Answering Dataset v2 (SQuAD v2), leveraging its character-level answer span annotations for 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*[1]) triple. We denote this dataset as SQuAD-PosQ. To analyze position 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], where all intervals are inclusive. We choose fixed-width intervals of 100 characters to enable fine-grained comparison of retrieval performance across different passage regions, while ensuring sufficient examples in each bucket. Each question is treated as a query, with its gold passage designated as the relevant target in a passage ranking task over the full retrieval corpus. This setup enables us to assess how the position of relevant content affects retrieval accuracy under

[1]Refers to the index position of the first character of the answer, with the length calculated in units of characters.

[2]The maximum observed index is 3120.

realistic conditions. A consistent performance drop for questions where the answer appears later in the passage would indicate the presence of position bias. For efficiency,, we additionally construct a smaller subset, SQuAD-PosQ-Tiny, consisting of 10,000 randomly sampled triplets, while keeping the retrieval corpus unchanged.

3.2 Generating Position-sensitive Questions

While SQuAD-PosQ serves as a useful benchmark to analyze position bias, 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 (Chen et al., 2024; Lee et al., 2025), raising concerns about evaluation leakage. To address these issues, we construct a synthetic dataset using passages from the FineWeb-edu, a large-scale, high-quality educational web text corpus. We sample 13,902 passages from the collection whose lengths range from 500 to 1,024 words. We instruct `gpt-4o-mini` (OpenAI, 2024a) to generate questions anchored to localized chunks of each passage, following carefully designed prompts (see Appendix B). Each passage is divided into three equal-length segments—*beginning*, *middle*, and *end*—and each question is assigned to one of these buckets based on the position of its supporting chunk. Additionally, we apply a two-stage filtering pipeline to ensure high-quality question generation by validating positional relevance and minimizing false negatives (see Appendix A.2). The resulting dataset, FineWeb-PosQ, contains 25,775 synthetic questions and facilitates rigorous evaluation of position sensitivity in longer-context retrieval. For efficiency, we also create a smaller version, FineWeb-PosQ-Tiny, by sampling 1,000 questions from each position category, resulting in a total of 3,000 questions.

Appendix A provides more details on dataset statistics, construction methodology, and empirical validation of the sampled subsets as reliable proxies for evaluating position bias.

3.3 Position Sensitivity Index

To quantify a retrieval model's sensitivity to the position of relevant content (i.e., position bias), we introduce a simple and intuitive metric called the **Position Sensitivity Index (PSI)**. This metric captures the model's worst-case performance degradation across different positional buckets. Given a set of position-specific evaluation scores $\mathbf{s} = \{s_1, \ldots, s_k\}$ (e.g., NDCG@10 for each position group), we define PSI as:

$$\text{PSI} = 1 - \frac{\min(\mathbf{s})}{\max(\mathbf{s})}, \quad \text{where } \max(\mathbf{s}) > 0. \quad (1)$$

Intuitively, PSI measures the relative drop from the best-performing position to the worst-performing one. A lower PSI suggests that the model's performance is more consistent across positional buckets, indicating reduced sensitivity to the location of relevant content within the passage. For example, if the scores are identical across all positions, we have $\min = \max$, resulting in $\text{PSI} = 0$, which signifies complete positional robustness. Conversely, a large gap between $\min$ and $\max$ pushes PSI closer to 1, signaling strong position bias. Compared to alternative dispersion metrics such as standard deviation or the coefficient of variation (CV) (Arachchige et al., 2022), PSI provides a worst-case perspective by explicitly quantifying the maximum relative degradation across positions. Note that the PSI formulation (Equation 1) is *scale-invariant* in that it captures only the relative variation across positions, independent of the absolute retrieval quality. However, this also means that PSI alone does not reflect a model's effectiveness. For instance, a model with uniformly low scores (e.g., all NDCG@10 values at 0.1) will have $\text{PSI} = 0$ despite being practically ineffective. Therefore, PSI should always be interpreted in conjunction with a measure of overall quality, such as the mean NDCG score across positions, to ensure that both robustness and retrieval performance are properly assessed.

4 Experiments

4.1 Experimental Setup

We perform a comprehensive evaluation across the full IR pipeline to assess the extent and impact of position bias, covering four distinct categories of retrieval models.

- **Sparse Retrievers**: `BM25` (Robertson et al., 1994)
- **Dense Retrievers**: `bge-m3-dense`[3] (Chen et al., 2024), `stella_en_400M_v5` (Zhang et al., 2025a), `text-embedding-3-large` (OpenAI, 2024b), `voyage-3-large` (VoyageAI, 2025), `jina-embeddings-v4` (Günther et al.,

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

Retrieval Models	SQuAD-PosQ							FineWeb-PosQ			
	0+	100+	200+	300+	400+	500+	*PSI* ↓	begin	middle	end	*PSI* ↓
Sparse Retrievers											
BM25	76.62	79.37	80.61	81.06	81.43	79.49	0.059	89.40	90.80	88.36	0.027
Dense Embedding-based Retrievers											
bge-m3-dense*	84.47	83.03	81.47	79.95	77.98	74.61	0.117	88.77	78.39	71.88	0.190
stella_en_400M_v5*	85.78	83.62	82.24	80.34	78.96	75.69	0.118	86.10	77.92	69.41	0.194
Qwen3-Embedding-0.6B*	82.60	81.93	79.08	77.36	75.39	71.48	0.135	88.54	78.83	65.61	0.259
text-embedding-3-large*	85.19	82.45	80.32	77.84	75.27	71.10	0.165	81.72	75.95	79.50	0.071
voyage-3-large*	89.93	89.32	89.17	88.70	88.09	86.73	0.036	92.76	87.46	83.38	0.101
jina-embeddings-v4*	82.50	80.55	78.87	77.33	75.91	72.94	0.116	88.35	77.46	69.80	0.210
Qwen3-Embedding-4B*	86.36	85.92	85.17	83.77	82.09	78.85	0.087	89.74	81.48	70.72	0.212
gte-Qwen2-7B-instruct*	85.13	83.85	83.33	81.71	80.13	77.75	0.087	84.24	79.07	75.90	0.099
NV-embed-v2*	93.04	93.55	93.48	93.02	92.48	90.72	0.030	77.24	85.12	85.98	0.102
Qwen3-Embedding-8B*	89.16	87.55	85.90	84.05	82.13	78.82	0.116	90.80	83.35	73.66	0.189
ColBERT-style Late-interaction Models											
colbertv2.0*	91.85	90.27	91.74	89.64	86.71	84.57	0.079	-	-	-	-
jina-colbert-v2*	93.52	92.42	93.28	92.58	91.80	78.14	0.164	91.69	56.45	45.91	0.499
bge-m3-colbert*	89.88	88.09	88.84	87.68	86.72	86.36	0.039	92.77	86.38	81.82	0.118
Full-interaction Reranker Models											
bge-reranker-v2-m3	93.53	93.56	94.69	94.50	94.42	94.52	0.012	94.25	96.10	94.87	0.019
Qwen3-Reranker-0.6B	92.11	91.43	91.53	91.65	90.60	89.69	0.026	95.03	94.97	92.46	0.027
gte-multilingual-reranker	90.70	91.10	92.59	91.84	91.57	92.03	0.020	94.70	95.73	95.51	0.011
Qwen3-Reranker-4B	93.32	92.84	93.38	93.94	92.57	93.26	0.015	95.06	96.58	95.23	0.016
bge-reranker-v2-gemma	94.31	94.01	94.73	94.80	94.55	94.55	0.008	94.38	95.84	96.02	0.017
Qwen3-Reranker-8B	93.38	93.48	93.81	94.20	93.83	94.31	0.010	95.61	97.02	96.74	0.015

Table 1: NDCG@10 scores ↑ and Position Sensitivity Index (PSI) ↓ of retrieval models on SQuAD-PosQ and FineWeb-PosQ. Models exhibiting notable position bias (i.e., PSI ≥ 0.03 on both datasets) are marked with *.

2025), `gte-Qwen2-7B-instruct` (Li et al., 2023b), `NV-embed-v2` (Lee et al., 2025), `Qwen3-Embedding-0.6B/4B/8B` (Zhang et al., 2025b)

- **ColBERT-style Late-interaction Models**: `colbertv2.0` (Santhanam et al., 2022), `bge-m3-colbert`[4] (Chen et al., 2024), `jina-colbert-v2` (Jha et al., 2024)
- **Full-interaction Reranker Models**: `bge-reranker-v2-m3` (Chen et al., 2024), `gte-multilingual-reranker-base` (Zhang et al., 2024), `bge-reranker-v2-gemma` (Li et al., 2023a), `Qwen3-Reranker-0.6B/4B/8B` (Zhang et al., 2025b)

We adopt NDCG@10 as our primary evaluation metric, which captures both retrieval accuracy and ranking quality within the top-10 retrieved results. To further quantify worst-case performance variations with respect to the position of relevant content, we introduce the Position Sensitivity Index (PSI) (see Section 3.3) as a complementary diagnostic metric. BM25 and dense embedding models are evaluated on the full datasets, whereas the more computationally 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.[5]

4.2 Experimental Results

4.2.1 BM25: Naturally Position-Robust

BM25, a classical sparse retrieval method based on term-matching, exhibits strong robustness to position bias across both SQuAD-PosQ and FineWeb-PosQ. Its NDCG@10 scores remain relatively stable across all positional buckets, with low PSI values of 0.059 and 0.027, respectively. This aligns with expectations: BM25 does not encode word order or any positional information, relying solely on keyword overlap. While this limits its ability to capture deeper semantic relationships,

[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.

[5] Due to its maximum sequence length of 512 tokens, `colbertv2.0` is incompatible with the longer-passage setting of FineWeb-PosQ, and is therefore excluded from evaluation on this dataset.

such position-agnostic behavior proves advantageous in scenarios where relevant content appears later in the passage. BM25 thus serves as a robustness baseline, demonstrating that retrieval quality need not necessarily deteriorate with content position.

4.2.2 Embedding Models: Widespread Bias

A wide range of dense embedding-based retrievers, including both open-source models (e.g., `bge-m3-dense`) and commercial offerings (e.g., `text-embedding-3-large`), exhibit substantial performance degradation as relevant content appears later in the passage. These results align with the head-position bias observed in prior work (Coelho et al., 2024; Fayyaz et al., 2025). Interestingly, the persistence of position bias appears unrelated to model size: from `Qwen3-Embedding-0.6B` to `Qwen3-Embedding-8B`, PSI remains consistently high despite increasing model capacity. Notably, `voyage-3-large` shows a much higher PSI on FINEWEB-POSQ (0.101) than on SQUAD-POSQ (0.036), suggesting potential evaluation leakage in widely used datasets like SQuAD, and underscoring the diagnostic value of the newly constructed FINEWEB-POSQ benchmark in revealing latent position bias. An unexpected case is `NV-embed-v2`, which displays a reversed trend on FINEWEB-POSQ: its lowest NDCG@10 score occurs at the beginning of passages. We leave the investigation of this reversal to future work, as it may be attributed to specific architectural design or distributional characteristics of the training corpus.

4.2.3 ColBERT-style Models: Persistent Bias

ColBERT-style late-interaction models balance retrieval efficiency and effectiveness by independently encoding queries and passages into multi-vector representations, followed by token-level interactions at inference time. Although they sometimes outperform dense retrievers in absolute NDCG@10, they still exhibit considerable position bias, especially on longer passages. For example, `jina-colbert-v2` suffers a sharp performance drop on FINEWEB-POSQ, from 91.69 (beginning) to just 45.91 (end), resulting in a PSI of 0.499—among the highest in our evaluation. This suggests that late interaction alone cannot fully compensate for position bias introduced during early-stage encoding. However, variation within the ColBERT family is noteworthy: `bge-m3-colbert` shows a much lower PSI than `jina-colbert-v2` on both datasets. Interestingly, under the same base encoder and training data, `bge-m3-colbert` clearly outperforms its dense counterpart, `bge-m3-dense`. This supports the idea that ColBERT-style training may help mitigate position bias, though it does not fully eliminate it.

4.2.4 Reranker Models: Effective Mitigation

Full-interaction reranker models, which apply deep cross-attention between query and passage, demonstrate the highest resilience to position bias among all model classes evaluated. All reranker models maintain consistently high NDCG@10 scores across positional buckets, with PSI values uniformly below 0.03. For instance, `bge-reranker-v2-m3` achieves NDCG@10 scores ranging from 93.53 to 94.69 on SQUAD-POSQ (PSI 0.012), and from 94.25 to 96.10 on FINEWEB-POSQ (PSI 0.019), indicating a high degree of robustness to the position of relevant content. These results underscore the strength of full cross-attention, which enables the model to flexibly attend to relevant spans regardless of position. From a system design perspective, these findings highlight that although dense embedding-based and ColBERT-style retrievers are vulnerable to head-position bias, incorporating an interaction-based reranking stage can substantially mitigate it. In high-stakes retrieval settings such as RAG applications, integrating a reranker serves as a crucial safeguard, ensuring that relevant information is accurately recognized and appropriately prioritized in the final ranking. However, this effectiveness relies on the assumption that relevant passages appear in the Top-K retrieval pool, underscoring the importance of the choice of K in practical deployments.

5 Conclusion

We conduct a comprehensive study of position bias in the modern IR pipeline. To enable realistic evaluation, we introduce two position-aware retrieval benchmarks: SQUAD-POSQ and FINEWEB-POSQ, repurposed from existing datasets while preserving semantic integrity. We further propose the Position Sensitivity Index (PSI), a simple and intuitive metric for quantifying position bias across retrieval models. Our findings reveal that while position bias primarily arises in embedding-based retrievers, it can be substantially mitigated by downstream interaction-based reranker models.

Limitations

This work has several limitations that open avenues for future research. First, our study focuses exclusively on position bias in English text retrieval, and the findings may not directly generalize to multilingual, cross-lingual, or even multimodal retrieval settings. Understanding how position bias manifests in such settings is an important next step. To this end, we are constructing a highly fine-grained and comprehensive position-aware retrieval benchmark, named POSIR[6], which spans multiple domains and languages, with potential extensions to image modalities, aiming to lay a solid foundation for future research on position bias. Second, our analysis does not yet provide a theoretical account of why embedding-based retrievers exhibit uneven information distribution in their vector representations. Without such a mechanistic understanding, it is difficult to design principled methods for mitigating position bias. Future work will explore connections to representation theory with the aim of developing more robust and unbiased text representation learning methods. Finally, our study abstracts away from user interaction effects. In realistic scenarios where multiple relevant passages exist, position bias may interact with human reading or clicking behavior: users tend to notice and rely on information presented earlier, while equally valid content appearing later may be overlooked. Investigating this interaction would yield a more comprehensive understanding of the practical implications of position bias in retrieval systems.

References

Chandima N. P. G. Arachchige, Luke A. Prendergast, and Robert G. Staudte. 2022. Robust analogs to the coefficient of variation. *Journal of Applied Statistics*, 49(2):268–290. PMID: 35707217.

Jianlyu Chen, Nan Wang, Chaofan Li, Bo Wang, Shitao Xiao, Han Xiao, Hao Liao, Defu Lian, and Zheng Liu. 2025. AIR-bench: Automated heterogeneous information retrieval benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19991–20022, Vienna, Austria. Association for Computational Linguistics.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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. Association for Computational Linguistics.

Mohsen Fayyaz, Ali Modarressi, Hinrich Schuetze, and Nanyun Peng. 2025. Collapse of dense retrievers: Short, early, and literal biases outranking factual evidence. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9136–9152, Vienna, Austria. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink training of bert rerankers in multi-stage retrieval pipeline. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 – April 1, 2021, Proceedings, Part II*, page 280–286, Berlin, Heidelberg. Springer-Verlag.

Michael Günther, Saba Sturua, Mohammad Kalim Akram, Isabelle Mohr, Andrei Ungureanu, Bo Wang, Sedigheh Eslami, Scott Martens, Maximilian Werk, Nan Wang, and Han Xiao. 2025. jina-embeddings-v4: Universal embeddings for multimodal multilingual retrieval. *Preprint*, arXiv:2506.18902.

Sebastian Hofstätter, Aldo Lipani, Sophia Althammer, Markus Zlabinger, and Allan Hanbury. 2021. Mitigating the position bias of transformer models in passage re-ranking. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I*, volume 12656 of *Lecture Notes in Computer Science*, pages 238–253. Springer.

Rohan Jha, Bo Wang, Michael Günther, Georgios Mastrapas, Saba Sturua, Isabelle Mohr, Andreas Koukounas, Mohammad Kalim Wang, Nan Wang, and Han Xiao. 2024. Jina-ColBERT-v2: A general-purpose

[6] https://huggingface.co/datasets/infgrad/PosIR-Benchmark-v1

multilingual late interaction retriever. In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 159–166, Miami, Florida, USA. Association for Computational Linguistics.

Zhiying Jiang, Raphael Tang, Ji Xin, and Jimmy Lin. 2021. How does BERT rerank passages? an attribution analysis with information bottlenecks. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 496–509, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

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.

Sean MacAvaney, Sergey Feldman, Nazli Goharian, Doug Downey, and Arman Cohan. 2022. ABNIRML: Analyzing the behavior of neural IR models. *Transactions of the Association for Computational Linguistics*, 10:224–239.

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*.

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.

David Rau, Mostafa Dehghani, and Jaap Kamps. 2024. Revisiting bag of words document representations for efficient ranking with transformers. *ACM Trans. Inf. Syst.*, 42(5).

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.

Yumeng Wang, Lijun Lyu, and Avishek Anand. 2022. Bert rankers are brittle: A study using adversarial document perturbations. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '22, page 115–120, New York, NY, USA. Association for Computing Machinery.

Xinyi Wu, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. 2025. On the emergence of position bias in transformers. In *Forty-second International Conference on Machine Learning*.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: BERT and beyond. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, pages 1–4, Online. Association for Computational Linguistics.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. 2025a. 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.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Zhenyu Zhang, Runjin Chen, Shiwei Liu, Zhewei Yao, Olatunji Ruwase, Beidi Chen, Xiaoxia Wu, and Zhangyang Wang. 2025c. Found in the middle: how language models use long contexts better via plug-and-play positional encoding. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

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 Dataset Cards

Table 2 presents summary statistics for the SQuAD-PosQ and FineWeb-PosQ datasets. Note that the two datasets differ in design: SQuAD-PosQ provides a fine-grained character-level positional analysis, while FineWeb-PosQ is constructed with a coarse-grained chunk-based segmentation. These differing granularities serve complementary purposes in analyzing positional effects across diverse settings. Some examples of FineWeb-PosQ are shown in Table 3.

A.1 Distribution Analysis of Answer Positions in SQuAD v2

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

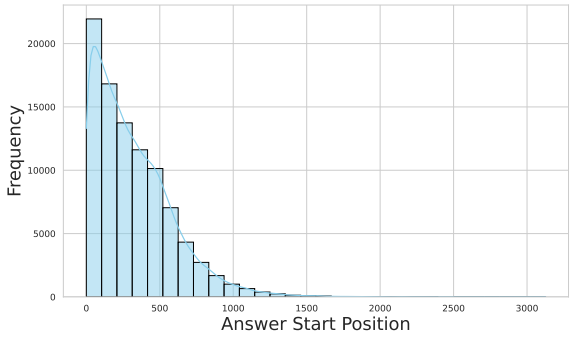

Figure 2: Distribution of answer start positions in SQuAD v2.

A.2 FineWeb-PosQ Construction Details

FineWeb-PosQ is built from the FineWeb-edu corpus, with the goal of creating a position-aware retrieval benchmark grounded in long, high-quality educational passages. We begin by selecting 13,902 passages between 500 and 1,024 words to ensure sufficient content length. Each passage is globally summarized using `gpt-4o-mini`, and then split into 256-word chunks using the `RecursiveCharacterTextSplitter`[7]. Each chunk, along with the global summary, is used to generate a (*question*, *answer*, *question_type*) triplet using `gpt-4o-mini`. Initially, we experimented with generating only questions, but found that approximately 40% of them were either unanswerable or misaligned with the source content. To address this, we adopt joint question–answer

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

	SQuAD-PosQ	*-Tiny	FineWeb-PosQ	*-Tiny
# Query	92,749	10,000	25,775	3,000
Mean Query Length	10.09	10.08	13.98	14.05
Std Query Length	3.56	3.56	4.01	4.11
# Passage	20,233	–	13,902	–
Min Passage Length	20	–	500	–
Max Passage Length	653	–	1,023	–
Mean Passage Length	117.19	–	710.79	–
Std Passage Length	50.22	–	132.34	–
Positional Bucket				
0+: [0–100]	21,220	2,252	beginning: 8,467	beginning: 1,000
100+: [100–200]	16,527	1,813		
200+: [200–300]	13,667	1,444	middle: 8,213	middle: 1,000
300+: [300–400]	11,514	1,210		
400+: [400–500]	10,089	1,108	end: 9,095	end: 1,000
500+: [500–3120]	20,384	2,237		

Table 2: Statistics of the SQuAD-PosQ and FineWeb-PosQ datasets. The two datasets use different bucketing schemes due to differences in construction methodology.

No.	Question	Position Tag
1	What is the purpose of the computerized vest developed by researchers at Georgia Tech?	beginning
2	What doctrine did John Wycliffe dispute, antagonizing the orthodox Church?	middle
3	What was the date of George IV's coronation?	end

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

generation, which substantially improved the quality, answerability, and relevance of the questions produced. To encourage diversity, we tag each question with a complexity label (*simple* or *complicated*), but all valid samples are retained regardless of complexity type. Each passage is divided into three equal-length regions: `beginning`, `middle`, and `end`. Each chunk is assigned to a span using a simple rule (Algorithm 1). In ambiguous cases where a chunk overlaps with two spans, we assign them to `middle` span for consistency.

To ensure the dataset is both positionally accurate and high-quality, we implement a two-stage filtering pipeline:

Stage 1: Validating Positional Relevance. The initial generation yields 265,865 question–chunk pairs across the 13,902 passages, with 15,961 from the beginning span, 199,742 from the middle span, and 50,162 from the end span. To ensure each question truly pertains to its labeled position span, we apply a consistency check using three reranker models, including `bce-reranker-base_v1`, `mmarco-mMiniLMv2-L12-H384-v1`, and `jina-reranker-v1-turbo-en`. Each model scores the question against the three segments (*beginning*, *middle*, *end*) of its corresponding passage. Only questions for which all three models agree that the labeled span yields the highest relevance score are retained. This aggressive filtering reduces the dataset to 117,008 questions—13,061 from the beginning span, 65,941 from the middle span, and 38,006 from the end span—ensuring the evaluation set is both positionally precise and semantically reliable. To further balance the dataset, we downsample each category to 13,061 questions—the size of the smallest category (beginning span)—yielding a total of 39,183 samples. Then, we verify fine-grained alignment using `DeepSeek-V3-0324`, prompting it to assign a relevance score (0–4) between the question and each segment (see prompt in Appendix B.3). A question is retained only if: (1) The score for its

labeled span is 3 or 4. (2) Its score is at least one level higher than any other span. This LLM-based filtering step takes 36 hours and results in 26,356 questions, with 8,658 from the beginning span, 8,433 from the middle span, and 9,265 from the end span.

Stage 2: Minimizing False Negatives. To reduce false negatives (relevant passages incorrectly regarded as irrelevant), we follow the three-step approach from Chen et al. (2025). *(1) Recall with Embedding Models.* For each question q_i, we use `jina-embedding-v3` to retrieve the top-1,000 relevant passages from the corpus (denoted $L_{\text{recall}} = \{p_1, \ldots, p_{1000}\}$). *(2) Pre-label with Rerankers.* We rerank L_{recall} using three rerankers: `jina-reranker-v1-turbo-en`, `bge-reranker-v2-minicpm-layerwise`, and `gte-reranker-modernbert-base`. A passage p_j is labeled positive by model M if its normalized score $r_j(M) \geq 0.5$. If a majority of the three models label d_j as positive, we pre-label it as positive; otherwise, negative. This step identifies 854 potential false negatives. *(3) Label with LLMs.* We further verify these potential false negatives using three LLMs: `deepseek-chat`, `gemini-2.5-flash`, and `gpt-4.1-mini`. Each LLM scores passage relevance from 0–4 (consistent with stage 1). A passage is retained as false negative only if at least two LLMs assign a score ≥ 3. This confirms 661 high-confidence false negatives. Given that these high-confidence items affect fewer than 3% of questions, we remove all associated questions to ensure data purity. We do not relabel passages to avoid introducing ambiguity in downstream evaluation.

After the above two filtering stages, the final dataset contains:

- **Questions:** 25,775
- **Passages:** 13,902
- **Position Distribution:**
  - Beginning: 8,467
  - Middle: 8,213
  - End: 9,095

Algorithm 1 POSITION TAGGING

Require: Total length z, chunk start index m, end index n
Ensure: Return tag: `beginning`, `middle`, `end`

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

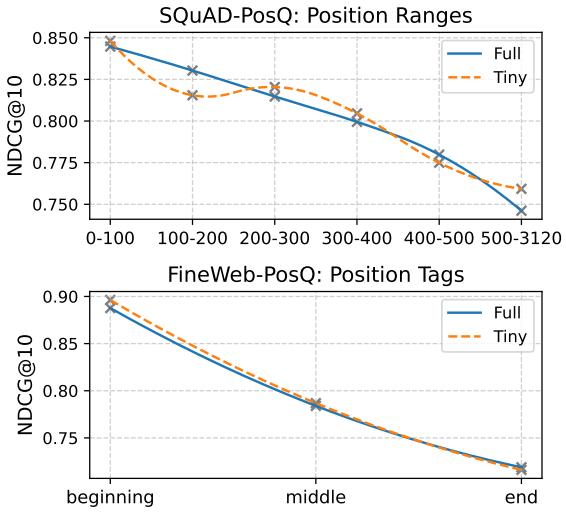

Figure 3: NDCG@10 scores of `bge-m3-dense` on Full vs. Tiny Datasets.

A.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. Figure 3 shows that `bge-m3-dense` achieves highly consistent NDCG@10 performance between the full and sampled datasets, particularly for FINEWEB-POSQ. These results confirm the feasibility of using the sampled subset to accelerate evaluation for computationally intensive models. Additionally, the experiments reveal a pronounced head-bias in `bge-m3-dense`, indicating a tendency to overly prioritize the beginning context while neglecting the middle and end segments during retrieval.

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

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.

A.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, 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 all models, reinforcing the observation that embedding models exhibit a strong position bias—favoring the beginning of the input while underrepresenting its later parts.

B Prompts

B.1 Prompt for Summarization

<task>
Given a passage, 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 passage, but do not exceed 400 words.
- You MUST only output the paraphrase, and do not output anything else.
</requirements>

<passage> {TEXT} </passage>

B.2 Prompt for Question Generation

<task>
Given a summary and a chunk of passage, 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>

Your output should be a JSON List:

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

B.3 Prompt for Relevance Estimation

Evaluate the relevance between the provided query and passage on a scale of 0-4, where:
0 = Completely irrelevant
1 = Slightly relevant (minimal connection)
2 = Moderately relevant (partial match)
3 = Highly relevant (covers most aspects)
4 = Perfectly relevant (document fully addresses query)

Query: {query}

Passage: {passage}

Output only a single integer from 0,1,2,3,4 without any additional text, explanations, or formatting. Higher values indicate stronger relevance.