Situated Embedding Models for Context-Aware Dense Retrieval

Junjie Wu[1], **Jiangnan Li**[2], **Yuqing Li**[3], **Lemao Liu**[4], **Liyan Xu**[2], **Jiwei Li**[5]
Dit-Yan Yeung[1], **Jie Zhou**[2], **Mo Yu**[2*]
[1]HKUST [2]Tencent [3]IIE-CAS [4]Fudan University [5]Zhejiang University
junjie.wu@connect.ust.hk {jiangnanli,moyumyu}@tencent.com

Abstract

Retrieval-augmented generation (RAG) over long documents typically involves splitting the text into smaller chunks, which serve as the basic units for retrieval. However, due to dependencies across the original document, contextual information is often essential for accurately interpreting each chunk. To address this, prior work has explored encoding longer context windows to produce embeddings for longer chunks, yet their gains in retrieval and downstream tasks remain limited. This is because (1) longer chunks strain the capacity of embedding models due to the increased amount of information they must encode, and (2) many real-world applications still require returning localized evidence due to constraints on model or human bandwidth. To this end, we propose an alternative approach to this challenge by representing short chunks in a way that is conditioned on a broader context window to enhance retrieval performance – *i.e.*, **situating** a chunk's meaning within its context. We further show that existing embedding models are not well-equipped to encode such situated context effectively, and thus introduce a new training paradigm and develop the first *situated embedding model*. To evaluate our method, we curate a book-plot retrieval dataset specifically designed to assess situated retrieval capabilities. On this benchmark, our 1B-parameter model substantially outperforms state-of-the-art embedding models, including several with up to 7B parameters.[1]

1 Introduction

Text embedding models (Wang et al., 2024a; Sturua et al., 2024; Nussbaum et al., 2024) encode textual inputs into vector spaces. These models enable efficient semantic representation and matching, thus are foundational to many applications involving retrieval-augmented generation (RAG) (Lewis et al., 2020), such as code generation (Wang et al., 2024b; Miao et al., 2024) and personal AI assistants (Martin and Johnson, 2023).

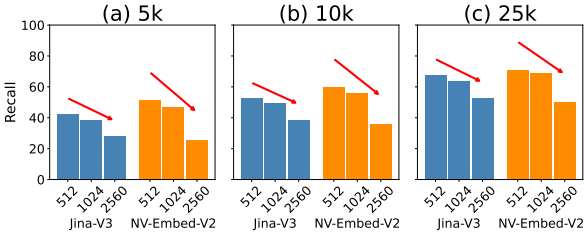

Figure 1: Comparison of the same embedding models that return the same lengths of texts with different chunk sizes on our evaluation task (§3). X-axis refers to chunk sizes. For example, when the return text is 5k and the chunk size is 1,024, the retriever returns top-5 chunks.

In these tasks, candidate documents are typically segmented into smaller chunks to facilitate efficient processing. However, since documents often exhibit a narrative or logical flow, the meaning of each chunk is highly dependent on its surrounding context. This highlights the need for text embeddings that capture broader contextual information to enable **context-aware retrieval**.

One straightforward approach to this issue is to increase chunk size, allowing each chunk to capture more information. This has motivated a wave of recent work on supporting long input sequences in embedding models, either by designing efficient bidirectional models (Chen et al., 2024; Sturua et al., 2024; Nussbaum et al., 2024), or by repurposing powerful unidirectional pre-trained LLMs as embedding generators (Li et al., 2023; Wang et al., 2024a; Moreira et al., 2024; Kim et al., 2024). These models can produce embeddings for sequences of up to 8,192 tokens or more.

However, it is often observed that simply *enabling longer input windows does not necessarily lead to better embeddings*. A key reason lies in the **limited capacity** of embedding vectors – embedding models must compress the information in the input text into a single vector. Intuitively, the

*Corresponding Author.
[1]https://huggingface.co/SituatedEmbedding

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 37–49

longer the input chunk, the more information it contains, and the more long-range dependencies across arbitrary pairs of chunks within a document it needs to capture. This increases the likelihood of critical information loss during compression. Existing models are trained by merely extending the context window, without explicitly learning how to represent such distributed contextual relationships, which leads to a counterintuitive outcome: applications built on long-chunk embeddings often underperform those using short-chunk embeddings, despite the latter discarding more contextual information. Figure 1 illustrates this effect on a book plot retrieval task (Xu et al., 2024a). When the same total length of text (5k/10k/25k tokens) is retrieved using the same embedding model (Jina or NV-Embed), recall consistently decreases as the documents are segmented into longer chunks.

Given the above challenges, we propose an alternative approach to context-aware embedding: directly incorporating the broader context surrounding each short chunk into its embedding. This enables the model to account for how the chunk is situated within the original document, resulting in more contextually informed representations. In other words, we aim to situate a chunk's meaning within its broader context (Yu et al., 2023; Xu et al., 2024a) during the embedding process—a strategy we refer to as **situated embedding**. By doing so, we alleviate the issue of capacity limitations: during encoding, the model only needs to identify and integrate context that is relevant to the target chunk, which is a more tractable task than modeling all dependencies across an extended input window.

Building on the definition of situated embedding, we first investigate whether existing embedding models can effectively generate situated embeddings. *Surprisingly, we find that this cannot be achieved simply by prompting existing embedding models*, as demonstrated in §4. To address this limitation, we develop the first dedicated situated embedding model specifically designed to handle this scenario. We achieve this through two techniques: (1) *Constructing context-dependent training instances* using publicly available user-annotated book notes. Platforms such as Douban[2] allow users to write notes anchored to particular book segments. We treat the note as a query and the anchor text as groundtruth, framing a retrieval task with ~1.6M query-candidate pairs. As user notes typically reflect the contextual understanding of surrounding context, it makes context-aware embeddings beneficial for this retrieval task. (2) *Promoting context usage through residual learning*. In many cases, a chunk alone may offer partial (usually ambiguous) clues about its relevance to the query, allowing models to exploit shortcuts. To counter this, we employ a residual architecture where the situated embedding model is trained to resolve the residual from a baseline chunk-only embedding model. This encourages the model to focus on the additional contextual information.

To evaluate models' context-aware embedding capability, we curate a book-plot retrieval task following (Xu et al., 2024a), which has been verified by previous work containing 7 books for its guaranteed requirement of context-aware embedding capability. Experiments demonstrate our situated embedding model's superior performance over all the state-of-the-art embedding models, including those with up to 7B parameters and with massive pre-training. Finally, we illustrate the generalizability of the trained models, through experimenting on three downstream tasks §6.

2 Our Situated Embedding Model

Training Data Construction We collect the notes and their associated anchor texts for ~100 most popular books according to Douban. We treat each user note as a query and its corresponding sentence in the book as a chunk, resulting in 1,614,007 query–chunk pairs. We reserve all the query–chunk pairs from our evaluation books and randomly select 1000 pairs for early-stopping.

Given a query–chunk pair, we define the situated context of the chunk as *a sequence of its surrounding sentences, including the chunk itself*. Specifically, we use the user-underlined texts anchored to their query as the chunk's situated context, as these texts naturally align with our definition. Due to variations in user behavior, the lengths of these situated contexts range from 37 tokens to several thousand, making our trained model robust to a wide range of context lengths.

After this process, each chunk will be contained within one such segment, and we regard the segment as the situated context of the chunk.

Residual Learning to Promote Situated Context Usage Prior studies, such as (Ettinger, 2020), have shown that BERT-based models often rely on shallow heuristics or partial, ambiguous clues when

[2] https://book.douban.com/

matching texts. This behavior hinders the model's ability to fully comprehend the entire input, which potentially explains why existing embedding models struggle to utilize long contextual information. To address this limitation, we adopt a residual learning framework (He et al., 2016), in which a situated embedding model is trained to resolve the residual from a baseline chunk-only embedding model, thereby equipping the trained model with a deeper understanding of situated context.

Specifically, we maintain two models, a baseline model Θ^b that embeds the chunk only and a situated model Θ^s that embeds the chunk situated within the context. For each query-chunk pair in the training data, we treat the chunk as the positive sample, and randomly sample 10 other chunks from the remaining chapters of the same book as negative samples. A query is embedded as $\tilde{\mathbf{q}} = \mathbf{q}^b + \mathbf{q}^s$, where $\mathbf{q}^b$ and $\mathbf{q}^s$ are embedding vectors from Θ^b and Θ^s, respectively. Similarly, a chunk is embedded as $\tilde{\mathbf{c}} = \mathbf{c}^b + \mathbf{c}^s$. The training loss on each query-chunk pair can then be defined as:

$$\mathcal{L}(\Theta^b, \Theta^s) = \frac{1}{N} \sum_{i=1}^{N=10} \max\big(0,\ \gamma + \mathrm{sim}(\tilde{\mathbf{q}}_j, \tilde{\mathbf{c}}^-_{j,i}) - \mathrm{sim}(\tilde{\mathbf{q}}_j, \tilde{\mathbf{c}}^+_j)\big), \quad (1)$$

where i is the index of negative chunk. See Appendix B for details of the training process.

3 Evaluation Dataset

Xu et al. (2024a) repurpose instances from the PlotRetrieval dataset (Xu et al., 2024b) to support the task of contextual retrieval. Their work focuses on a single book, demonstrating that incorporating a graph-based representation of the book can improve local chunk retrieval. This finding highlights that the plot retrieval task inherently requires situated understanding and retrieval capabilities.

Following their work, we repurpose the PlotRetrieval dataset into a chunk-level retrieval task *Book Plot Retrieval*. Specifically, we filter books in PlotRetrieval that are too short (i.e., $\leq$100,000 tokens), as they can typically be processed in a single input window and thus diminish the utility of RAG. We also exclude books with too few user notes, as well as less popular versions on the reading platform, which tend to have less diverse note styles.

This filtering process results in 7 evaluation books containing 1,394 diverse queries, which together constitute the *Book Plot Retrieval* task. The names of these books, along with the corresponding numbers of queries and candidate chunks, are summarized in Table 1. Note that for some English books, the PlotRetrieval dataset includes multiple Chinese translation versions, treating each version as a distinct book. We adopt the same setting in *Book Plot Retrieval* and denote the three translation versions of *Notre-Dame de Paris* in the 7 selected books as v1, v2, and v3, respectively. Among them, NDP-v1 is the version used in Table 2.

Book	Queries	Candidates
Notre-Dame de Paris (NDP)-v1	510	1288
Notre-Dame de Paris (NDP)-v2	153	1369
Notre-Dame de Paris (NDP)-v3	146	1347
Crime and Punishment (C&P)	134	1639
The Adventures of Tom Sawye (TATS)	173	154
The Red and the Black (TRB)	144	1294
Tess of the d'Urbervilles (TDU)	134	1093

Table 1: Statistics of books in the *Book Plot Retrieval* task.

When constructing the situated context for each chunk in the *Book Plot Retrieval* task, we first partition the chunk's corresponding book into segments of $[128, 384]$ tokens. We then sequentially group consecutive segments surrounding each chunk until the total length reaches a context limit that is uniformly sampled from [2,048, 6,144] tokens.[3] This grouped context serves as each chunk's situated context. During evaluation, we report Recall@10, Recall@20, and Recall@50 as the primary metrics.

4 Study I: Analysis of Existing Models on Generating Situated Embeddings

As the first step, we investigate the necessity of training a situated embedding model. That is, *are existing long-context embedding models capable of generating good situated embeddings?*

Setup. We investigate this question on the NDP-v1 book in our evaluation dataset. We compare the following models[4]: 1) *Long-context BERT models,* including BGE-M3 (Chen et al., 2024) and Jina-v3 (`Jina-Embeddings-v3`) (Sturua et al., 2024). 2) *LLM-based embedding models*: E5-Mistral (`E5-Mistral-7b-Instruct`) (Wang et al., 2024a), GTE-Qwen2 (`GTE-Qwen2-7b-Instruct`) (Li et al., 2023), and NV-Embed-v2 (Lee et al., 2024). 3) *Our trained situated model* (**Sit-M3**) from §2. Check Appendix D for additional details on model usages.

[3] See Appendix A for how we fix situated contexts' lengths.

[4] The models are selected based on their strong performance on the MTEB benchmark (Muennighoff et al., 2023).

Model	Size	Chunk-Only			+ Situated Context			+ Situated Summ.		
		@10	@20	@50	@10	@20	@50	@10	@20	@50
M3	0.5B	42.55	53.33	66.51	9.48	15.68	24.70	41.83	53.94	67.78
Jina-v3	0.5B	42.65	52.67	67.92	34.10	44.27	58.33	45.48	56.12	69.90
E5-Mistral	7B	43.18	51.93	66.65	14.77	24.49	32.68	44.58	54.32	68.27
GTE-Qwen2	7B	46.19	55.79	71.15	19.01	29.77	49.34	42.44	49.88	60.85
NV-Embed-v2	7B	**51.38**	60.11	71.16	21.01	30.20	42.18	49.25	58.33	70.85
Sit-M3 (Ours)	1B	51.06	**60.57**	**73.77**	**51.73**	**61.56**	**75.00**	**51.85**	**62.66**	**76.26**

Table 2: Recall results on NDP-v1. The maximum length is set to 8,192. Best results of each setting are boldfaced.

Setting	Model	Size	Recall		
			@10	@20	@50
Chunk-only	M3 (out-of-box)	0.5B	32.92	41.46	55.85
	M3 (trained)	0.5B	42.87	52.91	66.01
	Res-M3 (trained)	1B	43.43	51.74	65.51
Situated	Sit-M3 (Ours)	1B	**44.71**	**56.16**	**69.53**
	- Residual	0.5B	43.85	54.98	68.93

Table 3: Averaged results on the book plot retrieval task. Check Table 6 for full results on individual books.

Results. Table 2 presents the evaluation results, from which we draw the following conclusions:

• *Existing models do not have zero-shot situated embedding capability.* When enhancing the contexts to chunks, the performance of all the existing models degrades significantly (*i.e.*, comparing columns of *+Situated Context* and *Chunk-Only*), while the length of the situated context is well within their claimed maximum context window sizes. In contrast, our situated embedding model can effectively leverage contextual information, and largely surpasses the much larger 7B baselines.

• *The poor results partly sourced from limitation in understanding long inputs.* The failure of producing situated embeddings is partly from the existing models' (actual) insufficiency of handling long inputs. To see this, we in addition compare with the **LLM-generated situated summaries** approach (Anthropic, 2024), which prompts an LLM to generate a concise summary that reflects how a chunk is situated within its broader context as the contextual information. We ask GPT-4o (OpenAI, 2024) to generate the situated summaries and use them in the same way like the situated contexts. Note that we use this setting only for reference, because it does not make a fair comparison due to the involvement of a much stronger model in the pipeline with high computational cost.

From the results, all the models suffer from a much smaller degradation when using the summaries instead of situated contexts, while M3, Jina and E5 have their results slightly increased, reflecting that the baselines fail to situate the target chunk within long contexts. In contrast, our model achieves performance boost for both types of contextual inputs.

5 Study II: Contextual Retrieval

We evaluate our situated embedding model on the full 7 books of the book plot retrieval task to assess its effectiveness in enhancing contextual retrieval. As shown in Table 3, incorporating contextual information through situated embeddings significantly improves performance – our model, Sit-M3, consistently outperforms chunk-only baselines.

To ensure that the gains are not merely due to increased model capacity, we also train the same residual architecture on two chunk-only M3 models (Res-M3). It fails to yields improvements over the trained-M3 baseline, indicating that the advantage of our method primarily come from the effective use of contextual information. In addition, training without the residual architecture (- Residual) leads to degraded performance compared to our full Sit-M3, further supporting our training design.

6 Study III: Downstream Applications

Finally, we assess the generalizability of our situated embedding model on downstream applications that are not explicitly designed for contextual retrieval and contain only a limited portion of context-dependent examples. We evaluate on three tasks: (1) the *Recap Snippet Identification task* (Li et al., 2024), which aims to identify recap passages for a given paragraph; (2) *LoCoV1* (Saad-Falcon et al., 2024), a long-context retrieval task that requires processing inputs beyond simple chunking; (3) *LongStoryQA-large* (Qiu et al., 2024), a question-answering benchmark that requires processing in-

	Recap			LoCoV1	LSQA
Model	R@5	P@5	F1@5	nDCG@10	Ans-F1
M3 (trained)	30.52	49.81	37.44	98.77	52.24
Sit-M3 (Ours)	**31.21**	**51.11**	**38.33**	**99.38**	**53.91**

Table 4: Results on the Recap, LoCoV1, and LongStoryQA tasks. Check Appendix E for details and full results of the Recap and the LoCoV1 tasks.

puts exceeding the length limits of many LLMs.

We compare our trained M3 models with and without situated embedding capability. For Recap and LSQA, we retrieve the top-5 passages. For the QA task, we use GPT-4o to generate answers. Table 4 lists the results, showing that the situated embedding model consistently yields higher-quality retrieval, leading to improved results on both tasks.

7 Conclusion

This paper introduces the first situated embedding model designed to incorporate a chunk's contextual information directly into its embedding, enabling a deeper understanding of the chunk itself. Experimental results across multiple long-context understanding tasks demonstrate that situated embedding provides an effective alternative approach to contextual retrieval, and our proposed model serves as a strong first step in advancing this direction.

Limitations

This work is the first to train a situated embedding model, aiming to highlight the limitations of simply increasing input window size and to propose an alternative solution. From a modeling perspective, however, there are several limitations:

First, our training data is primarily sourced from the Chinese Internet, making the current model suitable only for Chinese-language tasks. Extending this approach to multilingual settings – similar to recent efforts such as (Hu et al., 2025) – is an important direction for future work.

Second, we validate our approach using BGE-M3, a 0.5B-parameter model. The same method can be scaled to larger models (e.g., 7B), where causal masking in LLMs naturally supports residual learning within a single model architecture. We leave this exploration to future work.

Acknowledgment

This work has been made possible by a Research Impact Fund project (RIF R6003-21) and a General Research Fund project (GRF 16203224) funded by the Research Grants Council (RGC) of the Hong Kong Government.

References

Anthropic. 2024. Enhancing rag with contextual retrieval.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Michael Günther, Isabelle Mohr, Daniel James Williams, Bo Wang, and Han Xiao. 2024. Late chunking: contextual chunk embeddings using long-context embedding models. *arXiv preprint arXiv:2409.04701*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Xinshuo Hu, Zifei Shan, Xinping Zhao, Zetian Sun, Zhenyu Liu, Dongfang Li, Shaolin Ye, Xinyuan Wei, Qian Chen, Baotian Hu, and 1 others. 2025. Kalm-embedding: Superior training data brings a stronger embedding model. *arXiv preprint arXiv:2501.01028*.

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. One thousand and one pairs: A "novel" challenge for long-context language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*.

Jihoon Kwon Sangmo Gu Yejin Kim, Minkyung Cho Jy-yong Sohn Chanyeol, Choi Junseong Kim, and Seolhwa Lee. 2024. Linq-embed-mistral: Elevating text retrieval with improved gpt data through task-specific control and quality refinement. linq ai research blog.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

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*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Jiangnan Li, Qiujing Wang, Liyan Xu, Wenjie Pang, Mo Yu, Zheng Lin, Weiping Wang, and Jie Zhou. 2024. Previously on the stories: Recap snippet identification for story reading. *arXiv preprint arXiv:2402.07271*.

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*.

Raiza Martin and Steven Johnson. 2023. Introducing notebooklm.

Jing Miao, Charat Thongprayoon, Supawadee Suppadungsuk, Oscar A Garcia Valencia, and Wisit Cheungpasitporn. 2024. Integrating retrieval-augmented generation with large language models in nephrology: advancing practical applications. *Medicina*, 60(3):445.

Gabriel de Souza P Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. 2024. Nv-retriever: Improving text embedding models with effective hard-negative mining. *arXiv preprint arXiv:2407.15831*.

Xiangyang Mou, Chenghao Yang, Mo Yu, Bingsheng Yao, Xiaoxiao Guo, Saloni Potdar, and Hui Su. 2021. Narrative question answering with cutting-edge open-domain qa techniques: A comprehensive study. *Transactions of the Association for Computational Linguistics*, 9:1032–1046.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037.

Zach Nussbaum, John X Morris, Brandon Duderstadt, and Andriy Mulyar. 2024. Nomic embed: Training a reproducible long context text embedder. *arXiv preprint arXiv:2402.01613*.

OpenAI. 2024. Hello gpt-4o.

Zexuan Qiu, Jingjing Li, Shijue Huang, Xiaoqi Jiao, Wanjun Zhong, and Irwin King. 2024. Clongeval: A chinese benchmark for evaluating long-context large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3985–4004.

Jon Saad-Falcon, Daniel Y Fu, Simran Arora, Neel Guha, and Christopher Ré. 2024. Benchmarking and building long-context retrieval models with loco and m2-bert. In *Proceedings of the 41st International Conference on Machine Learning*, pages 42918–42946.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Andreas Koukounas, Nan Wang, and Han Xiao. 2024. jina-embeddings-v3: Multilingual embeddings with task lora. *Preprint*, arXiv:2409.10173.

Voyage-AI. 2025. Introducing voyage-context-3: focused chunk-level details with global document context. Blog post.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024a. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916.

Zora Zhiruo Wang, Akari Asai, Xinyan Velocity Yu, Frank F Xu, Yiqing Xie, Graham Neubig, and Daniel Fried. 2024b. Coderag-bench: Can retrieval augment code generation? *arXiv preprint arXiv:2406.14497*.

Liyan Xu, Jiangnan Li, Mo Yu, and Jie Zhou. 2024a. Fine-grained modeling of narrative context: A coherence perspective via retrospective questions. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5822–5838.

Shicheng Xu, Liang Pang, Jiangnan Li, Mo Yu, Fandong Meng, Huawei Shen, Xueqi Cheng, and Jie Zhou. 2024b. Plot retrieval as an assessment of abstract semantic association. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 146–161.

Zhe Xu, Jiasheng Ye, Xiaoran Liu, Xiangyang Liu, Tianxiang Sun, Zhigeng Liu, Qipeng Guo, Linlin Li, Qun Liu, Xuanjing Huang, and Xipeng Qiu. 2025. DetectiveQA: Evaluating long-context reasoning on detective novels. In *Workshop on Reasoning and Planning for Large Language Models*.

Mo Yu, Jiangnan Li, Shunyu Yao, Wenjie Pang, Xiaochen Zhou, Zhou Xiao, Fandong Meng, and Jie Zhou. 2023. Personality understanding of fictional characters during book reading. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14784–14802.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*.

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. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

A Selection of Context Length for Building Situated Context of Chunks During Evaluation

In this section, we describe the selection process for the length of situated context for each chunk. As described in §3, we partition each chunk's corresponding book into segments by sequentially grouping consecutive sentences, and put these segments together to construct situated context. Therefore, to determine an appropriate situated context length, we conduct a preliminary experiment on the NDP-v1 book used in Table 2, varying the number of sentences included per chunk and evaluating the resulting recall scores using our trained Sit-M3 (ours) model. The experimental results are presented in Table 5.

As shown, setting the situated context length within the range of [2048, 6144] yields the best performance across various recall metrics. Consequently, we adopt this length range in all subsequent experiments.

Situated Context Length	Recall		
	@10	@20	@50
[512, 1536]	51.63	61.22	74.48
[1024, 3072]	**52.29**	<u>61.23</u>	74.15
[2048, 6144]	<u>51.73</u>	**61.56**	<u>75.00</u>
[4096, 12288]	50.62	59.11	**75.16**
[8192, 24576]	50.52	59.11	74.51

Table 5: Recall results of Sit-M3 (trained) on NDP-v1 using various lengths of situated context for chunks. The listed lengths correspond to multiples (4/8/16/32/64) of the average sentence range observed in books from the book plot retrieval task ([128, 384] tokens). The best results are **boldfaced** and the second best results are <u>underlined</u>.

B Additional Training Details of Our Situated Embedding Model

In this section, we describe additional details on how we attempt to train the first situated embedding model.

B.1 Model Initialization

Before the residual learning process described in §2, we initialize two models, Θ^b and Θ^s. While Θ^s is directly initialized from the BGE-M3 embedding model, we perform a prior training step on Θ^b to facilitate more effective residual learning.

Specifically, we initialize Θ^b from the same BGE-M3 embedding model as Θ^s. For each query–chunk pair in the training data, we treat the chunk as the positive sample and randomly sample 10 negative chunks from other chapters of the same book. We then obtain the query embedding $\mathbf{q}^b$ and chunk embedding $\mathbf{c}^b$ from Θ^b, and train Θ^b using the margin-based loss defined in Eq. 1, applied solely to this model. This prior training stage familiarizes Θ^b with the task of retrieving book chunks based on user notes, thereby providing a more informative foundation for the subsequent residual learning phase.

B.2 Training Configurations

All training procedures in this paper follow a consistent configuration. We use a learning rate of 2e-5 and a weight decay of 5e-2. The batch size is set to 80, and the maximum input sequence length is 8192 tokens, which corresponds to the input limit of BGE-M3. During training, we employ the development set introduced in §2 for early stopping. The model is evaluated on this set every 180 training steps, and training is terminated once both the training loss and development performance converge. The margin and temperature values used in the loss function are both set to 0.1. All experiments are conducted using two NVIDIA A100 GPUs.

C Full Results Decomposed to Books

The full results on individual books in *Book Plot Retrieval* are listed in Table 6.

D Details on Running Embedding Models

For all non-LLM embedding models (i.e., BGE-M3 and Jina-v3), we directly use the models to encode queries, chunks, and situated context, with the maximum input length set to 8192 tokens.

For E5-Mistral and GTE-Qwen2, we follow the official encoding guidelines provided at `https://huggingface.co/intfloat/e5-mistral-7b-instruct` and `https://huggingface.co/Alibaba-NLP/gte-Qwen2-7B-instruct`, respectively. In both cases, we prepend a one-sentence instruction to each query as required, as illustrated in Figure 2.

For NV-Embed-v2, we adopt the same input format as E5-Mistral and GTE-Qwen2 when encoding queries. For chunk encoding in the chunk-only setting of Table 2, we omit instructions, consistent with the E5-Mistral and GTE-Qwen2 setups. In all other settings in Table 2 where additional context is included, we follow the official prompt

Model	Book	@10	@20	@50
M3 (out-of-box)	NDP-v1	42.55	53.33	66.51
	NDP-v2	33.22	44.18	60.62
	NDP-v3	43.79	51.63	64.38
	C&P	21.64	24.63	45.15
	TATS	38.73	47.40	63.01
	TRB	23.61	34.72	47.22
	TDU	26.87	34.33	44.03
	Avg	32.92	41.46	55.85
M3 (trained)	NDP-v1	48.79	58.45	73.29
	NDP-v2	46.58	56.16	67.81
	NDP-v3	50.33	62.09	73.53
	C&P	36.57	42.91	54.85
	TATS	42.20	56.94	70.81
	TRB	36.80	43.06	58.33
	TDU	38.81	50.75	63.43
	Avg	42.87	52.91	66.01
Res-M3 (trained)	NDP-v1	49.30	58.73	71.93
	NDP-v2	48.29	54.79	66.78
	NDP-v3	50.33	59.15	72.22
	C&P	35.45	39.93	52.99
	TATS	45.66	56.36	72.25
	TRB	35.42	40.97	59.72
	TDU	39.55	52.24	62.69
	Avg	43.43	51.74	65.51
Sit-M3 (No Res)	NDP-v1	48.76	58.53	72.22
	NDP-v2	47.26	57.88	69.86
	NDP-v3	51.31	63.07	74.84
	C&P	37.31	45.90	57.09
	TATS	49.42	59.25	76.01
	TRB	33.33	47.22	64.58
	TDU	39.55	52.99	67.91
	Avg	43.85	54.98	68.93
Sit-M3 (Ours)	NDP-v1	51.73	61.56	75.00
	NDP-v2	49.66	59.25	71.92
	NDP-v3	50.65	66.67	74.84
	C&P	35.07	42.91	56.34
	TATS	47.98	60.98	76.88
	TRB	36.11	46.53	65.28
	TDU	41.79	55.22	66.42
	Avg	**44.71**	**56.16**	**69.53**

Table 6: Full results on book plot retrieval.

format of NV-Embed-v2 (https://huggingface.co/nvidia/NV-Embed-v2), as shown in Figure 3.

When running the latter two columns of experiments in Table 2, we append the situated context or situated summary to each chunk using the delimiters "</s>" and "\n\n", respectively. The concatenated sequence is then treated as a new chunk and encoded as a whole.

E Details and Full Results on the Recap and the LoCoV1 tasks

For the Recap tasks, we follow settings introduced in (Li et al., 2024) and run experiments on the DGSD and TCOMC sets. The full results are listed in Table 7.

Book	Model	R@5	P@5	F1@5
DGSD	M3 (trained)	31.82	41.31	35.95
	Sit-M3 (Ours)	**32.36**	**41.96**	**36.54**
TCOMC	M3 (trained)	29.22	58.31	38.93
	Sit-M3 (Ours)	**30.06**	**60.26**	**40.11**

Table 7: Full results on the Recap task of Table 4.

Model	nDCG@10
M2-BERT-8K	82.00
M2-BERT-32K	89.30
M3 (trained)	98.77
Sit-M3 (Ours)	**99.38**

Table 8: nDCG@10 results on the LoCoV1 task of Table 4. For M2-BERT-8K and M2-BERT-32K, we use the results reported in Table 13 of (Saad-Falcon et al., 2024). The best result is **boldfaced**.

For the LoCoV1 tasks, we conducted experiments using our model Sit-M3 (Ours), M3 (trained), and also the M2-BERT-8K/32K models proposed by (Saad-Falcon et al., 2024) on one of the non-saturated subsets of LoCoV1, *passage_retrieval*. When evaluating on LoCoV1, each passage was segmented into chunks based on the paragraph boundaries specified in the task. The overall passage score was computed as the maximum score among its constituent chunks. For Sit-M3 (Ours), we also constructed and incorporated situated context for each chunk during evaluation. The full results of the four models are listed in Table 8.

F Study IV: Significance Test

To further validate the reliability of our results, we have also performed significance test on our results in Table 2, Table 3, and Table 4.

For Table 2 and Table 3, we compare the performance of our best model, Sit-M3 (Ours), against the strongest baseline. Since recall scores are available for each query, we conduct paired t-test to determine whether the improvements are statistically significant. For Table 4, we similarly perform paired t-tests using the corresponding metric on each query. The results are shown below, where "**" means the bottom result is significant better than the upper one with $p < 0.01$, and "*" means the bottom result is significant better than the upper one with $p < 0.05$.

As shown in Table 9, Table 10, Table 11, Ta-

Instruct:
Given a user note query, retrieve the passages that are most relevant to the content or context described in the query.

Query:
{QUERY}

Figure 2: The query format of E5-Mistral and GTE-Qwen2

Your task is to embed passages for retrieval. Your input consists of the target passage and its context. You need to find relevant information from the context to enhance the target passage embedding such that it captures the meanings of the passages situated within the context.

context:
{CONTEXT}

passage:
{PASSAGE}

Figure 3: Prompt for NV-Embed-v2.

Model	Size	@10	@20	@50
NV-Embed-v2	7B	51.38	60.11	71.16
Sit-M3 (Ours)	1B	51.06	60.57	73.77*

Table 9: Significance test for the chunk-only setting on NDP-v1.

Model	Size	@10	@20	@50
Jina-v3	0.5B	34.10	44.27	58.33
Sit-M3 (Ours)	1B	51.73**	61.56**	75.00**

Table 10: Significance test for the +situated context setting on NDP-v1.

Model	Size	@10	@20	@50
NV-Embed-v2	7B	49.25	58.33	70.85
Sit-M3 (Ours)	1B	51.85*	62.66**	76.26**

Table 11: Significance test for the +situated summ. setting on NDP-v1.

Model	Size	@10	@20	@50
Res-M3 (trained)	1B	43.43	51.74	65.51
Sit-M3 (Ours)	1B	44.71*	56.16**	69.53**

Table 12: Significance test for Table 3.

ble 12, Table 13, in most cases our Sit-M3 model significantly outperforms the strongest baseline, further demonstrating its effectiveness.

	Recap			LoCoV1	LSQA
Model	R@5	P@5	F1@5	nDCG@10	Ans-F1
M3 (trained)	30.52	49.81	37.44	98.77	52.24
Sit-M3 (Ours)	**31.21**	**51.11***	**38.33**	**99.38***	**53.91***

Table 13: Significance test for Table 4.

G Study V: Training Situated Embedding with Additional Base Models

To further demonstrate the generalizability of our approach, we performed additional experiments using the latest Qwen3-Embedding-8B (Zhang et al., 2025) model as the base encoder. Leveraging the same training data described in §2, we fine-tuned Qwen3-Embedding-8B to produce situated embeddings. Thanks to its unidirectional decoder—where preceding tokens cannot attend to subsequent ones—we can concatenate each chunk with its surrounding context to simultaneously derive both the chunk-only and situated embeddings. This design eliminates the need to maintain two distinct models, as required by Sit-M3. We evaluate the resulting model, Sit-Qwen3 (Ours), on NDP-v1 following the setup in Table 2, and present the results in Table 14.

As can be seen, similar to Sit-M3, Sit-Qwen3 (Ours) effectively leverages contextual information and substantially outperforms all other baselines, including Qwen3-Embedding (8B). It also achieves

Model	Size	Chunk-Only			+ Situated Context			+ Situated Summ.		
		@10	@20	@50	@10	@20	@50	@10	@20	@50
M3	0.5B	42.55	53.33	66.51	9.48	15.68	24.70	41.83	53.94	67.78
Jina-v3	0.5B	42.65	52.67	67.92	34.10	44.27	58.33	45.48	56.12	69.90
E5-Mistral	7B	43.18	51.93	66.65	14.77	24.49	32.68	44.58	54.32	68.27
GTE-Qwen2	7B	46.19	55.79	71.15	19.01	29.77	49.34	42.44	49.88	60.85
NV-Embed-v2	7B	51.38	60.11	71.16	21.01	30.20	42.18	49.25	58.33	70.85
Qwen3-Embedding	8B	51.58	61.32	73.47	48.01	58.91	71.86	51.20	61.63	76.10
Sit-M3 (Ours)	1B	51.06	60.57	73.77	51.73	61.56	75.00	51.85	62.66	76.26
Sit-Qwen3 (Ours)	8B	**66.81**	**74.36**	**84.32**	**68.98**	**79.32**	**86.68**	**67.81**	**76.37**	**84.37**

Table 14: Recall results of models including Sit-Qwen3 (Ours) on NDP-v1. The maximum length is set to 8,192.

the best performance under the + *Situated Context setting*, highlighting its strong ability to exploit contextual information for improved retrieval. Furthermore, Sit-Qwen3 (Ours) surpasses Sit-M3, benefiting from its larger parameter size. Overall, these results demonstrate that our proposed method can be extended to various popular base embedding models, further validating its reliability.

H Study VI: Comparison of Context-aware Embedding Models

In addition to the situated embedding models proposed in this work, several existing approaches also attempt to integrate contextual information into the retrieval process. To further demonstrate the effectiveness of our situated embedding models compared to these methods, we select two representative and widely adopted baselines for comparison.

The first baseline is the widely used late-chunking method (Günther et al., 2024), which generates chunk embeddings by splitting them after encoding the entire context. For this approach, we evaluated `jina-embeddings-v3` by enabling its late-chunking mode through the official API (denoted as Jina-v3-late). Additionally, we included another recent context-aware closed-source embedding model, `voyage-context-3`, which claims to employ a novel context-aware mechanism during training. Since the model size of `voyage-context-3` is undisclosed, we further added Qwen3-Embedding-8B trained under our situated context setting as an additional baseline (Sit-Qwen3 (Ours) in Table 14). As both `jina-embeddings-v3` and `voyage-context-3` incorporate their own built-in contextual retrieval mechanisms, we followed their default configurations without introducing extra situated context. For Sit-M3 and Sit-Qwen3 (Ours), we report results under the situated context setup. All experiments were conducted on NDP-v1.

Model	Size	@10	@20	@50
Jina-v3-late	1B	45.68	56.29	70.30
Sit-M3 (Ours)	1B	51.73	61.56	75.00
voyage-context-3	unk	58.54	68.47	79.09
Sit-Qwen3 (Ours)	8B	**68.98**	**79.32**	**86.68**

Table 15: Recall results of different context-aware embedding models on NDP-v1.

As shown in Table 15, Sit-M3 (Ours) surpasses Jina-v3-late with a comparable model size, and Sit-Qwen3 (Ours) substantially outperforms both voyage-context-3 and jina-embeddings-v3 across the board. These results demonstrate that situated embedding models are highly effective at handling contextualized retrieval.

We attribute the stronger performance of our situated context setting to its being an easier task for base embedding models to learn. Intuitively, longer input chunks require capturing long-range dependencies across arbitrary chunk pairs within a document, which raises the risk of losing critical information during compression (lines 69–74, [1]). Such lost information is difficult to recover later, even with late-chunking. By contrast, our method explicitly incorporates how each chunk is situated within the original document, thus yielding more contextually informed representations.

I Study VII: Training Situated Embedding for Long-Context QA

In this section, we demonstrate that the concept of situated embeddings can be broadly applied to general-purpose benchmarks. Since question answering is widely regarded as a unified formulation of many NLP tasks, we evaluate our approach across a variety of long-context QA benchmarks.

Evaluation Tasks We evaluate on a variety of story understanding tasks that requires processing inputs exceeding the length limits of many LLMs,

including *NarrativeQA* (Kočiskỳ et al., 2018), the multichoice QA task from ∞*Bench* (Zhang et al., 2024), the newly release *DetectiveQA* (Xu et al., 2025) and the public subset of *NoCha* (Karpinska et al., 2024). These tasks cover different genres, both English and Chinese languages, and task types (*e.g.*, free-form QA, multi-choice QA and claim verification). We retrieve top-3/5/10 with the compared embedding models and use Qwen2.5-72B (4-bit quantized) model to generate the results.

Training Data Construction We follow Mou et al. (2021) and build retrieval training data based on NarrativeQA (Kočiskỳ et al., 2018). We then fine-tune the Qwen3-Embedding model to obtain our Sit-Qwen3 model, following the residual design described in Appendix G.

Overall Results Table 17 shows that our embedding model trained on QA data consistently outperforms its counterpart without situated embeddings. It also substantially outperforms the original Qwen3 embedding model, particularly on top-3 and top-5 results.

One notable observation regarding performance degradation with larger retrieved context on NoCha is that, once the key plot is retrieved, additional context tends to consist mainly of distractors, causing an LLM with weaker reasoning ability to lose focus. To verify this, we evaluated the advanced Gemini-2.5-Flash on the retrieved chunks by our situated embedding model, achieving top-3/5/10 pair accuracies of 55.6/57.1/57.1 without degradation. This confirms that the necessary evidence is saturated within the top-5 results.

Fine-Grained Evaluation on Retrieval Results Finally, we perform a fine-grained evaluation on the DetectiveQA dataset, which includes human-annotated evidence locations. Specifically, DetectiveQA provides two types of evidence annotations: *Answer Evidence*, which refers to the text span that directly yields the answer; and *Clue Evidence*, which refers to the supporting information that connects the evidence to the correct answer, mirroring the logical reasoning steps a detective would follow to solve the mystery. For reference, we also compare with the commercial late-chunking approach voyage-context-3 (Voyage-AI, 2025) in this setting.

Table 18 shows that our situated embedding model achieves a substantial improvement in answer-evidence recall over all other models, which directly contributes to its higher final-answer accuracy. Compared with the out-of-the-box Qwen3 model, it yields a 13–16% gain in answer recall and an overall 10% improvement in final-answer accuracy. Our model also demonstrates a clear advantage over the commercial voyage-context-3 model across most metrics, despite the latter benefiting from proprietary in-house training data.

Model	**Size**	**@10**	**@20**	**@50**
Qwen3	8B	51.58	61.32	73.47
Situated within Consecutive Contexts				
Sit-Qwen3 (Ours)	8B	68.98	79.32	86.68
Situated within UGC Contexts				
Sit-Qwen3 (Ours)	8B	**75.10**	**83.12**	**90.17**

Table 16: Recall results of different context-aware embedding models on NDP-v1.

J Study VIII: Embeddings Situated within Broader Types of Contexts

Finally, we show that the idea of situated embeddings is not limited to narrative contexts composed of consecutive chunks (**Consecutive Contexts**), but can also be applied to situate text within broader and more diverse types of contexts.

To this end, we provide each chunk with its associated user comments as additional context (**UGC Contexts**) and ask the embedding model to integrate useful information from these comments to enrich the chunk's semantics. Such user-generated content represents a distinct and complementary form of contextual information.

We conduct experiments on the NDP dataset. When constructing the contextual data, we use user comments from the different versions of the book, ensuring that the user notes used as queries and those used as context do not overlap and most likely come from different groups of readers (since few users read multiple versions of the same book and leave notes on each). As shown in Table 16, situating text within UGC contexts further enhances retrieval performance, yielding around a 20% improvement over the out-of-the-box Qwen3 model and achieving over 90% Recall@50. These results demonstrate that the concept of situated embeddings generalizes effectively across different types of contextual information, enabling stronger performance when the appropriate context is applied to a given scenario.

Model	NarrativeQA	∞Bench-En.MC	DetectiveQA	NoCha (Public)
	F1	Acc	Acc	Pair Acc
Qwen3 (out-of-box)	27.5/30.8/32.2	75.1/80.4/86.0	62.5/68.7/73.2	42.9/41.3/**46.0**
Qwen3 (trained on QA)	29.5/31.9/32.4	**83.0**/84.7/88.7	70.5/78.2/81.8	**54.0**/52.4/36.5
Sit-Qwen3 (trained on QA)	**31.1/32.0/34.4**	**83.0/86.9/90.0**	**73.2/78.7/82.3**	**54.0/55.6/46.0**

Table 17: Results on the story QA tasks. We report results with top-3/5/10 retrieved chunks.

Model	Answer Recall			Clue Recall			Final Accuracy		
	Top-3	Top-5	Top-10	Top-3	Top-5	Top-10	Top-3	Top-5	Top-10
voyage-context-3	36.1	46.8	63.3	**24.8**	33.8	48.1	68.7	73.5	79.8
Qwen3 (out-of-box)	29.6	37.8	55.5	23.8	31.9	46.5	62.5	68.7	73.2
Qwen3 (QA)	35.8	50.5	66.4	23.7	33.0	48.0	70.5	78.2	81.8
Sit-Qwen3 (trained on QA)	**42.5**	**54.5**	**69.3**	24.6	**34.0**	**49.2**	**73.2**	**78.7**	**82.3**

Table 18: Study on the effects of improved retrieval on the DetectiveQA dataset, which provides evidence passage annotations.