Context is Gold to find the Gold Passage: Evaluating and Training Contextual Document Embeddings

Anonymous ACL submission

Abstract

A limitation of modern document retrieval embedding methods is that they typically encode passages (chunks) from the same documents independently, often overlooking crucial contextual information from the rest of the document that could greatly improve individual chunk representations.

In this work, we introduce *ConTEB* (Context-aware Text Embedding Benchmark), a benchmark designed to evaluate retrieval models on their ability to leverage document-wide context. Our results show that state-of-the-art embedding models struggle in retrieval scenarios where context is required. To address this limitation, we propose *InSeNT* (In-sequence Negative Training), a novel contrastive post-training approach which combined with *late chunking* pooling enhances contextual representation learning while preserving computational efficiency. Our method significantly improves retrieval quality on *ConTEB* without sacrificing base model performance. We further find chunks embedded with our method are more robust to suboptimal chunking strategies and larger retrieval corpus sizes. We open-source all artifacts at `http://hf.co/<anonymous>`.

1 Introduction

The ability to rapidly process and query large-scale textual corpora is a cornerstone of many industrial applications, ranging from the analysis of medical records and legal briefs to large-scale administrative archives. As these collections grow in size and complexity, advanced approaches to information retrieval (IR) —particularly Retrieval-Augmented Generation (RAG) (Lewis et al., 2020)— have attracted widespread interest, yet, dealing with long documents remains an open challenge.

While long context encoders have been recently developed (Zhang et al., 2024; Warner et al., 2024a), along with long context embedding models (Zhu et al., 2024), modern document retrieval pipelines typically segment lengthy documents into smaller chunks to optimize the granularity for efficient retrieval and readability of the retrieved content (Liu, 2022; Xu et al., 2024; Jiang et al., 2024). Traditionally, these chunks are then *independently* fed to an embedding model, and stored in a vector database for efficient future query matching. By doing so, these systems remove strong semantic and conceptual links between the split passages, directly affecting the resulting representations. An example is illustrated in Figure 2: embedding the sentence *"He became emperor in 1804."* without leveraging information about the person at hand (*Napoléon*) given in previous paragraphs will make matching queries related to *Napoléon* difficult.

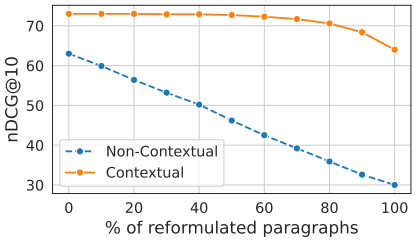

Figure 1: **Importance of Contextual Information:** Starting from a set of queries and mostly self-contained document paragraphs from the *Football*, we progressively reformulate paragraphs to remove information redundant with the rest of the document. This leads to sharp performance declines in standard retrieval approaches, but not in contextual retrieval approaches.

Recognizing the significant business value of incorporating broader contextual information into retrieval, major companies have explored leveraging large generative language models (LLMs) to mitigate this limitation. Some approaches attempt to circumvent retrieval altogether by feeding millions of tokens into the model's context window at runtime (Gemini Team et al., 2024), while others reformulate individual passages by concatenating them with document-level summaries and context (Anthropic, 2024). However, these methods are

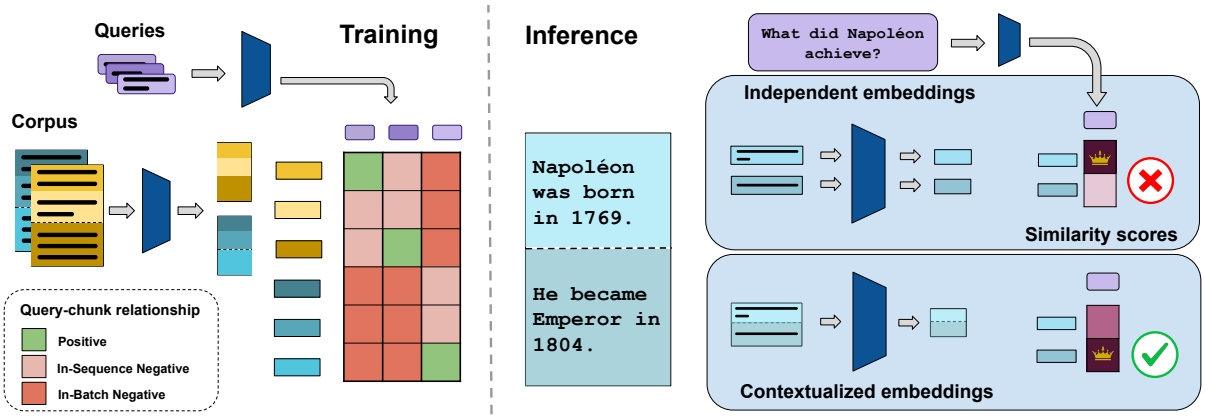

Figure 2: *Training (Left).* With respect to a single query, each chunk inside a batch plays a different role, depending on its original document, and the positive chunk. *Inference (Right).* Traditional embedding methods (top) produce embeddings that do not include potentially essential contextual information. Contextualized embeddings (bottom) can integrate document-wide information in individual chunk representations, augmenting embedding relevance and improving downstream retrieval performance.

prohibitively expensive at scale when dealing with corpora comprising thousands of documents.

Despite the critical importance of contextualized retrieval, standard benchmarks fail to capture this challenge. Evaluations traditionally focus on assessing the effectiveness of embedding models (Thakur et al., 2021; Muennighoff et al., 2022; Saad-Falcon et al., 2024), but they rely on datasets where document chunks are by design self-contained answer to the queries, which is a largely idealized scenario in practice (Thakur et al., 2025). Consequently, benchmarks fail to highlight the limitations of current retrieval strategies in handling context-dependent passages. Worse, recent findings by Zhou et al. (2025) indicate that some widely-used benchmarks exhibit biases that favor standard context-agnostic retrieval methods. Companies such as Anthropic have acknowledged these issues and maintain proprietary contextual retrieval benchmarks that remain unavailable to the public[1], underscoring the gap between academic evaluations and real-world industrial needs.

Contribution 1: *ConTEB*. We introduce the *Context-aware Text Embedding Benchmark*, designed to assess the ability of retrieval systems to leverage information from the entire document when indexing and retrieving document chunks. *ConTEB* comprises both custom-designed tasks for fine-grained analysis, and practical retrieval evaluation settings spanning multiple document types, domains, and situations in which leveraging context is helpful to produce more meaningful chunk representations. We evaluate standard embedding methods on the benchmark and find they struggle when contextual awareness is required.

Contribution 2: *Efficient Contextual Training*. Improving upon the *Late Chunking* method (Günther et al., 2024), we propose a novel embedding post-training method that optimizes information propagation between same-document chunks at indexing time to ensure embeddings are better contextualized. Our method largely boosts performance on *ConTEB*, with minimal computational overhead. Through extensive ablations, we detail critical design choices and show our method improves displays increased robustness to sub-optimal chunking strategies and produces representations that scale better with corpus size.

We open-source all project artifacts, including the benchmark, models and training data[2].

2 Problem Formulation & Related Work

2.1 Retrieval Frameworks

In this paper, we consider the traditional retrieval framework where a retrieval system given a query q, searches a corpus $\mathcal{D}$ for relevant documents. Each document $d \in \mathcal{D}$ is scored based on its content by first embedding the text into a vector space, and then computing a similarity measure. The similarity between a query q and a document d is defined as

$$\text{sim}(q, d) = f\big(\phi(q), \phi(d)\big)$$

where ϕ maps text into an n-dimensional vector space and $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a similarity function,

[1] https://www.anthropic.com/news/contextual-retrieval

[2] http://hf.co/<anonymous_for_peer_review>.

such as cosine similarity or dot product.

In applied settings, individual documents are often too long to be practical for retrieval purposes (Liu, 2022; Zhong et al., 2025a). Each document d is thus divided into segments called *chunks* by a partitioning function $\mathcal{P}$ defined as

$$\mathcal{P}(d) = \{c_1, c_2, \ldots, c_{N_d}\}$$

In the *standard retrieval* setting, the score is computed solely based on chunk content:

$$\text{sim}(q, c) = f\big(\phi(q), \phi(c)\big)$$

Additional information (priors) is however often available to the document embedding system. Typically, knowledge of the entire corpus $\mathcal{D}$, or of *structural metadata* M_c such as neighboring document chunks obtained through $\mathcal{P}$, can be leveraged by a modified embedding function ϕ_2, yielding the following similarity score:

$$\text{sim}(q, c) = f\Big(\phi(q), \phi_2\big(c, M_c, \mathcal{D}\big)\Big)$$

This work is centered on efficiently integrating priors about the entire document when embedding a sub-document chunk.

2.2 Integrating Contextual Information

Neural embedding models for passage-level text representation, popularized by SentenceBERT (Reimers and Gurevych, 2019), have enabled retrieval systems to move beyond lexical matching (Robertson et al., 1994). To include contextual information in these retrievers, previous works proposed methods that either operate *offline* during *indexing*, or online during *querying* when faced with a user request.

Indexing. The chunking strategy is a crucial design choice and often aims to optimize chunk self-containment. Fixed-size approaches with overlaps preserve continuity, while structure-aware chunking respects natural text boundaries, such as paragraphs or sentences. Semantic chunking, by contrast, splits text into topic-aligned segments. These methods appear in frameworks such as LlamaIndex (Liu, 2022) and LangChain (Chase, 2022), but different queries may need different chunk sizes. Thus, dynamic chunking techniques have emerged to adapt segmentation on the fly (Zhong et al., 2025b; Qian et al., 2024). Beyond optimizing chunking, some indexing approaches enrich chunks with broader context by prepending LLM-generated document summaries, contextual information or metadata Anthropic (2024); Poliakov and Shvai (2024). Similarly, Morris and Rush (2024) demonstrate that appending learned "corpus" embeddings to queries and documents can further improve retrieval. Other indexing-time techniques involve organizing chunks into higher-level data structures. For example, Edge et al. (2024) and Sarthi et al. (2024) cluster related chunks into semantic graphs or tree hierarchies.

Querying. In contrast, *query-time* solutions rely on iterative or agentic loops to refine retrieval dynamically. LLMs can be used to iteratively update the query or request additional chunks based on partial results (Xiong et al., 2021; Trivedi et al., 2023), or even to run "self-checks" and seek extra context when needed (Asai et al., 2023). While these adaptive techniques can better address complex, multi-hop queries, they typically require much more computational resources during inference.

3 *ConTEB*: Context-aware Text Embedding Benchmark

3.1 Benchmark Design

Existing benchmarks often rely on (or assume) self-contained document chunks. This creates a misleading perception that contextualization offers little to no benefit, which in practice is rarely the case. To address this gap, the *ConTEB* benchmark philosophy is to explicitly be composed of tasks in which leveraging document-wide context should lead to performance improvements. Our benchmark originates from two sources: new datasets specifically created for *ConTEB*, and repurposed academic datasets. We take special care in selecting data sources spanning from multiple domains, including realistic industrial scenarios.

Why Context? Context can help resolve ambiguity, such as distinguishing between multiple meanings of a word or resolving pronouns and entity references (co-reference resolution). It is also crucial when documents have a structured format, like legal or scientific texts, where understanding table of content hierarchy is key.

Concept. To isolate the importance of contextual cues and diminish other confounding factors, we construct three benchmark tasks to study contextualization in controlled experimental settings (Allen-Zhu, 2024). We also evaluate more practical retrieval settings at larger scale where we suspect

	Dataset	Queries	Docs	Tokens per Chunk	Chunks per Document	Context Utilization
In Domain	MLDR	100	100	170.5	15.4	Document-level reasoning
	NarrativeQA	8575	355	154.5	4.9	Document-level reasoning
	SQuAD	2067	2067	19.1	8.5	Chunk not self-contained
Out of Domain	**Football**	1767	224	86.1	3.6	Co-reference resolution
	Geography	5283	530	113.6	4.3	Co-reference resolution
	Insurance	120	1	80.7	60.0	Structure understanding
	Covid-QA	1111	115	153.9	29.1	Chunk not self-contained
	ESG Reports	36	30	205.5	123.4	Context disambiguation
	NanoBEIR*	650	56 723	199.4	1	No context is needed

Table 1: Merged *ConTEB* dataset details. Controlled datasets are highlighted in **bold blue**. NanoBEIR values are summed over the 13 datasets that compose it.

contextualization to help, and in which we rely on organic, pre-existing query-document pairs.

3.2 Benchmark Construction

Our generic benchmark curation pipeline is composed of three stages. We provide additional curation details in Appendix A.

1: Chunking. We select long documents spanning a variety of domains and chunk them through a structure-aware method[3] (Rajpurkar et al., 2016; Möller et al., 2020; Kočiský et al., 2017; Chen et al., 2024; Faysse et al., 2025).

2: Pairing. We use manual answer span annotations (*SQuAD*, *ESG*) or synthetically label them with a LLM (*CovidQA*, *MLDR*, *NarrativeQA*), to match queries with chunks obtained in Stage 1. This ensures queries are not solvable by design (Thakur et al., 2025). Alternatively, in our controlled experiment tasks, we generate queries pertaining to the chunks manually (*Insurance*) or synthetically using LLMs (*Football*, *Geography*).

3: Sabotage. The manually created questions in *Insurance* are designed to be ambiguous without prior knowledge of the document structure. This is manually verified in this phase. Going a step further, in *Football* and *Geography*, we reformulate chunks with the help of a LLM to remove explicit mentions of the original document's theme which all queries mention. We do so in all but the first chunks of each document, explicitly enforcing the need for context.

In addition to our contextual scenarios, we use *NanoBEIR* (Thakur et al., 2021) to evaluate non-regression on standard non-contextualized embedding tasks.

[3]RecursiveCharacterSplitter with a threshold of 1000 characters (Chase, 2022)

By combining hard tasks in controlled environments, repurposed academic benchmarks, and real-world industrial queries, our benchmark provides a comprehensive assessment of retrieval models in both standard and context-dependent retrieval scenarios.

3.3 Training Dataset

Open training data is a key factor to ensure fair comparison across methods and robust conclusion. In addition to our benchmark, we construct and release a training dataset composed of query and document chunk pairs. It includes the training splits of *MLDR* and *NarrativeQA*, repurposed with our previously detailed pipeline. To increase the number of queries, we further use *GPT-4o* to generate relevant supplementary synthetic queries. We also concatenate *SQuAD* chunks from the same Wikipedia article, keeping track of the original question-passage associations. The full dataset contains 9881 unique long documents (3698 tokens on average), corresponding to a total of 232'587 chunks and 307'241 queries (see subsection A.6). Scaling the dataset to more sources, through diverse synthetic augmentations and refinement–based augmentation methods (Lee et al., 2024; Wang et al., 2024) is left for future work.

3.4 Baselines

Training-Free. We evaluate a selection of off-the-shelf methods that are strong in their size categories such as a standard single-vector embedding model based on ModernBERT (`modernbert-embed-large` (Warner et al., 2024b; Chaffin, 2025b)), its multi-vector ColBERT equivalent (Khattab and Zaharia, 2020; Chaffin, 2025a) and *Okapi BM25* (Robertson et al., 1994), a strong

lexical matching method. Additionally, we compare against various contextualization approaches. Specifically, we include Anthropic's contextual retrieval approach (Anthropic, 2024)[4], and evaluate Late Chunking (Günther et al., 2024) without specific fine-tuning using `modernbert-embed-large`. These methods cover standard practices with varying level of complexities and indexing budgets.[5]

Training-Based. For fair evaluation, we also fine-tune the sentence embedding method `modernbert-embed-large` on the training dataset with the same batch construction strategy as when training our main method, ensuring performance differences only stem from methodological design.

4 Training Contextual Embedders

In this work, we leverage recent advances in long-context embedding models (Zhang et al., 2024; Warner et al., 2024a) to improve upon existing approaches through novel training strategies.

4.1 Architecture

Late Chunking. Late Chunking (Günther et al., 2024) (LC) is a training-free token pooling technique designed to enable information propagation across same-document chunks. Formally, given a document d split into chunks $\{c_1, \ldots, c_{N_d}\}$, dense retrievers compute independent representations:

$$\phi(d) = [\phi(c_1), \phi(c_2), \ldots, \phi(c_{N_d})]$$

In Late Chunking, chunks are concatenated and the whole sequence representation is computed in a single-forward pass:

$$H = \phi(c_1 \oplus c_2 \oplus \cdots \oplus c_{N_d})$$

where $H = [h_1, h_2, \ldots, h_T]$ consists of token-level representations. We then apply average pooling within each original chunk to obtain chunk-wise representations:

$$\phi_{LC}(c_i) = \frac{1}{|c_i|} \sum_{t \in c_i} h_t, \quad \forall i \in \{1, \ldots, N_d\}$$

This allows each chunk representation to benefit from contextualization over the full document before aggregation.

Late Interaction. Late Interaction (LI) models (Khattab and Zaharia, 2020; Chen et al., 2024) are retrieval methods that do not pool token representations and instead store all token embeddings of each document. This approach boosts performance, especially on long-context retrieval tasks (Warner et al., 2024a; Zhu et al., 2024), at the expense of storage cost. In this work, we propose extending Late Chunking approaches to LI models by applying standard LC but simply forgoing the final pooling and storing token embeddings depending on their original chunk memberships.

$$\phi_{LI}(c_i) = \{ h_t : t \in c_i \}, \quad \forall i \in \{1, \ldots, N_d\}$$

Setup. As the base single-vector embedding model for our experiments, we use `modernbert-embed-large` (Chaffin, 2025b) (396M parameters), which is fine-tuned for retrieval tasks using the method from Nussbaum et al. (2024). Respectively, we leverage `GTE-ModernColBERT` (Chaffin, 2025a) (149M parameters) for our late interaction experiments. Both models are based on ModernBERT (Warner et al., 2024a) which supports a context length of up to 8,192 tokens, significantly surpassing the 512-token limit of traditional BERT models, and thereby enabling the processing of longer documents in a memory efficient manner, which is critical to our method.

4.2 Learning Objective

Late Chunking enables information "leakage" between chunks of the same document. While this training-free method showed promises, we construct a learning objective to explicitly optimize contextual embedding models for this setting. Our aim is twofold: optimizing chunk representations to integrate relevant document-level information, all while ensuring they retain their specificity with respect to other same-document chunks, in order to prevent embedding collapse.

Previous works (Karpukhin et al., 2020; Ni et al., 2021; Izacard et al., 2021; Li et al., 2023; Wang et al., 2022; Nussbaum et al., 2025) have relied on various learning objectives inspired by the contrastive learning literature (Schroff et al., 2015). A natural choice is the InfoNCE objective (Oord et al., 2018), which samples "negative" embeddings from other documents of the same batch.

In our approach, we combine it with an auxiliary *in-sequence* contrastive loss, where chunks originating from the same document as the positive

[4]We use `Qwen-2.5-7B-Instruct` as the generative model which we serve on a 80GB A100 GPU with vLLM and `modernbert-embed-large` as the embedding model

[5]We also evaluate RAPTOR (Sarthi et al., 2024) with `Qwen-2.5-7B-Instruct` and `cde-small-v2` (Morris and Rush, 2024) but find them to be poorly adapted to our problem settings.

serve as hard negatives during training. Intuitively, training Late Chunking models contrastively with chunks from *different* documents encourages information propagation within each document and improves document identification. On the other hand, the contrastive term between same-document chunks ensures each chunk retains its specificity, and remains identifiable w.r.t. to its neighbors. This aspect is further motivated by the fact that in practice, queried corpora often contain negative documents stemming from the same source. Figure 2 illustrates chunk roles across a training batch.

Training Loss. To balance the contribution of in-sequence and in-batch negatives, we define the weighted InfoNCE loss as:

$$\mathcal{L} = \lambda_{\text{seq}}\mathcal{L}_{\text{seq}} + (1 - \lambda_{\text{seq}})\mathcal{L}_{\text{batch}} \quad (1)$$

where $\lambda_{\text{seq}} \in [0, 1]$. Loss terms are defined as:

$$\mathcal{L}_{\text{seq}} = -\mathbb{E}\left[\log \frac{\exp\left(q \cdot k^{+}/\tau\right)}{\sum_{k_i \in \mathcal{N}_{\text{seq}}} \exp\left(q \cdot k_i/\tau\right)}\right]$$

$$\mathcal{L}_{\text{batch}} = -\mathbb{E}\left[\log \frac{\exp\left(q \cdot k^{+}/\tau\right)}{\sum_{k_j \in \mathcal{N}_{\text{batch}} \cup \{k^{+}\}} \exp\left(q \cdot k_j/\tau\right)}\right]$$

Here, q denotes the query representation, and k^{+} is the gold chunk representation, which belongs to $\mathcal{N}_{\text{seq}}$, the set of chunks from the same sequence as k^{+}. Temperature $\tau > 0$, and $\mathcal{N}_{\text{batch}}$ is the set of all in-batch samples that do not belong to $\mathcal{N}_{\text{seq}}$. This extends to late interaction models by replacing the dot product between query and chunk embeddings by ColBERT's *MaxSim* between the multiple query and document token embeddings.

By tuning λ_{seq}, we can adjust the relative importance of in-sequence versus in-batch contrastive learning (Figure 3) resulting in our *InSeNT* method.

4.3 Model training

Our training strategy (*InSeNT*) is designed to be lightweight and to occur on top of capable pretrained embedding models without degrading their capabilities. We use AdamW, a cosine decay learning rate scheduler with a 5% warm-up phase and a learning rate of $5e-5$ and train for 2 epochs on our training dataset. Batches are constructed by sampling 4 long documents per device, retrieving all corresponding chunks and concatenating them with a separator token in between. As documents in our training set contain more than 20 chunks on average, which are themselves often linked to one or multiple queries, a batch contains more than 100 query, positive, negatives triplets to learn on.[6] A single epoch takes less than 1 H100 GPU hour.

5 Results

Document-wide context is essential. As seen in Table 2, methods leveraging contextual information widely outperform non-contextual methods across *ConTEB* tasks. These results highlight the critical role of context-aware embeddings in improving retrieval performance in such settings, whether through untrained late chunking approaches or expensive context-aware reformulation approaches. As expected, the gap is even more notable in *ConTEB*'s controlled setting experiments.

Improving contextual information propagation. Our results clearly show that *InSeNT* variants outperform their untrained counterpart (+14.6 nDCG@10 for ModernBERT, +11.5 for ModernColBERT). Importantly, this is not due to the nature of the training data itself; the non-contextual ModernBERT model trained on the same data (ModernBERT + Training) does not improve upon the untrained baseline. Furthermore, the tasks that display the biggest improvements are the controlled setting tasks *Insurance, Football*, that are explicitly designed to elicit information given in previous paragraphs, and that are out-of-domain w.r.t. our training set.

Late Interaction. Interestingly, while LI models are good at long-context retrieving, they are poorly suited to out-of-the-box late chunking (-0.3 nDCG@10 w.r.t. ModernColBERT without LI). We posit that since token embeddings are never pooled, these models learn very local features and cannot leverage information from neighboring tokens. Once trained with our method, ModernColBERT+*InSeNT* displays large performance gains across the board (+11.5 nDCG@10 w.r.t. ModernColBERT + Late Chunking), showcasing an increased ability to leverage external context.

Context can add noise. The *CovidQA* task sticks out from the rest as untrained late chunking approaches severely degrade performance. Qualitative analysis, as well as the strong performance of the non-contextualized ModernColBERT method, indicate that the query-chunk pairing are often very extractive and match on technical medical terms,

[6] In *MB+Training*, data is sampled the same way for fair evaluation but flattened in batch, corresponding to per-device batch sizes of more than 100.

	In-Domain					Out-Of-Domain					
	Practical Settings					Controlled Settings					Non-Contextual
	MLDR	SQuAD	NarrativeQA	COVID-QA	ESG Reports	Football	Geography	Insurance	Average	Runtime (ms/doc)	NanoBEIR
Non-Contextual Models											
BM25	69.4	56.2	74.7	53.7	19.9	12.2	45.6	0.0	41.5	**4.29**	43.4
ModernBERT Large	78.4	73.4	77.9	61.7	36.8	19.1	56.2	12.4	52.0	17.83	63.2
ModernColBERT	83.5	74.2	80.4	**78.2**	44.2	30.2	68.5	16.1	59.4	14.99	67.7
ModernBERT Large + Training	78.7	74.0	77.3	55.2	20.0	22.9	58.7	13.9	50.1	16.44	54.5
Untrained Contextual Models											
Anthropic Contextual	85.4	77.1	77.7	60.7	34.8	53.9	89.4	**100.0**	72.4	1890.94	63.2
ModernBERT Large + Late Chunking	78.5	77.1	75.8	40.0	31.7	54.6	89.6	41.0	61.0	15.81	63.2
ModernColBERT + Late Chunking	84.1	75.7	80.7	75.5	44.4	31.3	67.9	13.2	59.1	7.41	**68.2**
Trained Contextual Models											
ModernBERT Large + *InSeNT*	88.7	**80.9**	81.3	56.0	43.1	63.9	**90.7**	**100.0**	**75.6**	15.26	63.2
ModernColBERT + *InSeNT*	**90.1**	75.1	**83.5**	67.7	**48.3**	**64.6**	89.8	45.9	70.6	7.57	59.2

Table 2: Evaluation (nDCG@10) of baseline models and our proposed method on *ConTEB*. Runtime is per-document indexing time in milliseconds; smaller is better, so the fastest model is bolded.

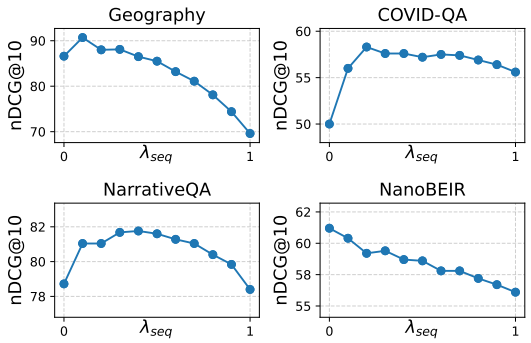

Figure 3: **Importance of λ_{seq}:** Results for ModernBERT-Large trained with varying λ_{seq}. Optimal values depend on the task, but integrating both in-sequence and in-batch negatives is crucial to performance.

thus rendering context less useful. Our results show that naively applying late chunking in this setting adds noise and leads to notable performance drops (-21 nDCG@10), which are in large part recovered through our training method (+16 nDCG@10).

λ_{seq} matters. The training objectives are to induce chunk representations to integrate document-level information (role of *in-batch* negatives) while maintaining their specificity with respect to other same document chunks (role of *in-sequence* negatives). By varying λ_{seq} from Equation 1, we weight the importance of both objectives.

After training a series of models with varying λ_{seq}, we see on Figure 3 that training with only in-sequence or in-batch negatives yields the worse results, and the optimal λ_{seq} varies depending on the task. When documents need to be disambiguated between one another (*NanoBEIR*, *Geography*), up-weighting in-batch negatives seems optimal. On tasks where the challenge lies in locating information within a given document (*NarrativeQA*, *Covid-QA*), in-sequence negatives play a large role, but still need to be combined to in-batch negatives. Striking the optimal trade-off is thus very use-case dependent, and we opt for $\lambda_{seq} = 0.1$ after tuning on the validation split of our training dataset.

Efficiency-Performance. As shown in the Runtime column of Table 2, our approach is very capable on contextual tasks, yet does not add much computational overhead. In fact, we find slight indexing speed improvements, attributed to our approach's reduced need for padding in-batch sequences of different lengths. While *Anthropic Contextual* achieves sensibly similar performances on *ConTEB*, it relies on costly LLM-based summarization and chunk reformulation, that are hardly scalable to huge corpora (120x slower).

Short-Context Performance. Careful hyperparameter tuning enables our best model to maintain strong performance on standard non-contextual benchmarks (*NanoBEIR*), demonstrating that long-context optimization does not compromise short-context retrieval. Interestingly, LI models suffer from more degradation, which we posit is due to the original reliance on very local features modified through our training. Mixing in non-contextual "replay" data during training or merging models

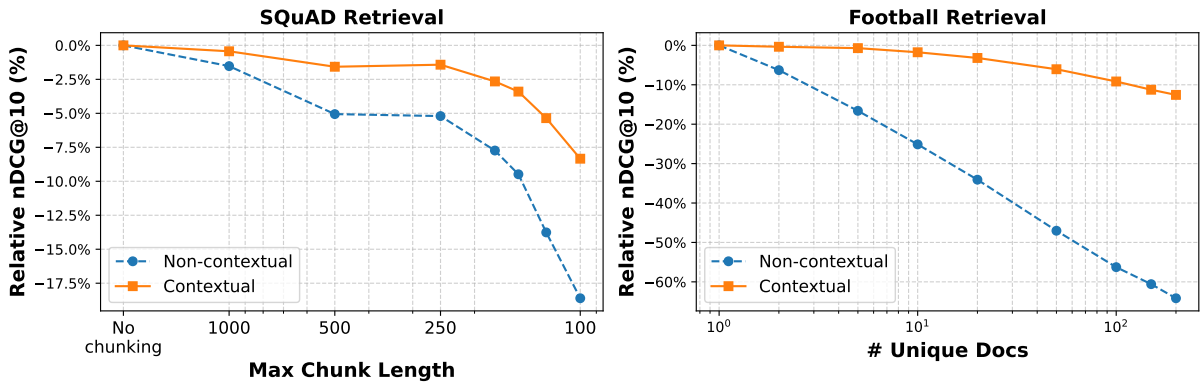

Figure 4: Contextualized models trained with InSeNT are more robust to aggressive chunking strategies that remove essential information from chunks (left), and scale better with corpus size and ambiguity (right).

(Wang et al., 2025) should further enable preserving the original embedding model's performances.

6 Ablations

Robustness to chunking. We assess our method's robustness to poor chunking strategies using *SQuAD* annotations. Each originally self-contained chunk is split in multiple progressively smaller sub-chunks to while we keep track of the annotated answer span to identify the gold chunk. Eventually, these sub-chunks become too small to be self-contained and end up lacking sufficient information to be relevantly embedded on their own. Figure 4 (left) demonstrates that contextual embeddings greatly improves robustness w.r.t. suboptimal chunking. The model is able to elicit information from neighboring chunks to integrate contextual information within smaller sub-chunks, leading to a much more uniform retrieval performance across a wide range of chunk sizes.

Robustness to corpus size. Common in the industry are templated documents that differ mostly by a key aspect (year, company name) but contain otherwise very similar information. We study the dynamics of retrieval performance w.r.t. to the amount of similar documents in the corpus by computing scaling laws in which we iteratively vary the number of unique documents (composed of multiple chunks) in the corpus. We observe in Figure 4 (right) that contextual embeddings scale vastly differently than their independently embedded counterpart. Intuitively, the greater the amount of similar documents and chunks in the corpus, the harder it is for a retrieval system to match the correct ones, but when embedding models are able to leverage external context, this effect is attenuated.

Information Propagation. We experiment with concatenating semantically similar yet independent short chunks as "artificial" long documents. The resulting model is contextual as it uses late chunking, but exhibits performances in-line with non-contextual baselines (*ModernBERT Large + Training*). We posit training on arbitrarily concatenated chunks, which by design are not contextually linked, teaches the model not to use information from neighboring chunks. This highlight the necessity of sourcing organic long-context data during training to induce correct training dynamics. Details in Table 5 in Appendix C.

7 Conclusions

In this work, we introduced *ConTEB*, a benchmark designed to assess the effectiveness of retrieval models in leveraging document-wide contextual information. Our evaluation demonstrates that standard retrieval models struggle in context-dependent settings, while our proposed approach *InSeNT*, which combines late chunking and a novel training methodology performs strongly on *ConTEB* without additional compute costs.

Future Work. Scaling our approach with recent decoder models with extended context lengths (e.g., 1M+ tokens (Yang et al., 2025)) would enable embedding entire books or lengthy documents in a single forward pass, potentially unlocking new capabilities for large-scale document retrieval. Second, adapting our method to multi-modal embedding pipelines that have less control over the chunking strategy (page level visual document embeddings for instance) could further enhance retrieval systems in industrial applications with visually rich contextual documents (Faysse et al., 2025; Ma et al., 2024).

Limitations

While our approach enhances retrieval performance in context-dependent settings, limitations persist.

Context Length. Our method is applied to long-context encoders that currently support sequences of up to 8k tokens. Scaling this approach to handle 1M+ token contexts with decoder-based models presents significant compute and memory challenges. Additionally, it requires rethinking data construction processes to ensure longer documents are effectively leveraged.

Data Generation. The creation of training and evaluation data relies on existing datasets and semi-synthetic generation pipelines. However, a fully automated and scalable method for generating high-quality queries that effectively induce non-trivial context utilization remains an open challenge.

Evaluation. While our model demonstrates strong cross-domain performance, further validation in real-world applications, various use cases, and multiple languages is necessary to assess its robustness and generalizability.

Ethical Considerations

Bias. As our method introduces a novel way of leveraging document-wide context, the nature of information propagation between chunks remains uncertain. This may introduce biases that traditional embedding models do not encounter, necessitating further analysis.

Ecological Impact. Our post-training approach is computationally efficient, with total training and evaluation runs requiring fewer than 100 GPU hours on H100 hardware. By providing a cost-effective alternative to LLM-dependent contextualization techniques, we aim to reduce the environmental footprint of large-scale retrieval systems.

Social Impact. Improved retrieval capabilities can drive significant business benefits, particularly in industries that rely on processing extensive and structured documents, such as legal, medical, and financial sectors.

References

Zeyuan Allen-Zhu. 2024. ICML 2024 Tutorial: Physics of Language Models. Project page: https://physics.allen-zhu.com/.

Anthropic. 2024. Introducing contextual retrieval. Accessed: 2025-02-10.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *Preprint*, arXiv:2310.11511.

Antoine Chaffin. 2025a. Gte-moderncolbert.

Antoine Chaffin. 2025b. Modernbert-embed-large.

Harrison Chase. 2022. LangChain.

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. *arXiv preprint*. Version Number: 3.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2025. Colpali: Efficient document retrieval with vision language models. *Preprint*, arXiv:2407.01449.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, and 1118 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

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

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised Dense Information Retrieval with Contrastive Learning. *arXiv preprint*. Version Number: 4.

Ziyan Jiang, Xueguang Ma, and Wenhu Chen. 2024. Longrag: Enhancing retrieval-augmented generation with long-context llms. *Preprint*, arXiv:2406.15319.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. *arXiv preprint*. Version Number: 3.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2017. The narrativeqa reading comprehension challenge. *Preprint*, arXiv:1712.07040.

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. *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, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv preprint*. Version Number: 4.

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

Jerry Liu. 2022. LlamaIndex.

Xueguang Ma, Sheng-Chieh Lin, Minghan Li, Wenhu Chen, and Jimmy Lin. 2024. Unifying multimodal retrieval via document screenshot embedding. *Preprint*, arXiv:2406.11251.

Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. 2020. COVID-QA: A question answering dataset for COVID-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, Online. Association for Computational Linguistics.

John X. Morris and Alexander M. Rush. 2024. Contextual document embeddings. *Preprint*, arXiv:2410.02525.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. MTEB: Massive Text Embedding Benchmark. *arXiv preprint*. Version Number: 3.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. 2021. Large dual encoders are generalizable retrievers. *Preprint*, arXiv:2112.07899.

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

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. 2025. Nomic embed: Training a reproducible long context text embedder. *Preprint*, arXiv:2402.01613.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *arXiv preprint*. Version Number: 2.

Mykhailo Poliakov and Nadiya Shvai. 2024. Multi-meta-rag: Improving rag for multi-hop queries using database filtering with llm-extracted metadata. *Preprint*, arXiv:2406.13213.

Hongjin Qian, Zheng Liu, Kelong Mao, Yujia Zhou, and Zhicheng Dou. 2024. Grounding language model with chunking-free in-context retrieval. *Preprint*, arXiv:2402.09760.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint*. Version Number: 1.

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).

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. *Preprint*, arXiv:2402.07440.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. *Preprint*, arXiv:2401.18059.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. Publisher: arXiv Version Number: 3.

Nandan Thakur, Jimmy Lin, Sam Havens, Michael Carbin, Omar Khattab, and Andrew Drozdov. 2025. Freshstack: Building realistic benchmarks for evaluating retrieval on technical documents. *Preprint*, arXiv:2504.13128.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. *arXiv preprint*. Version Number: 4.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *Preprint*, arXiv:2212.10509.

Ke Wang, Nikolaos Dimitriadis, Alessandro Favero, Guillermo Ortiz-Jimenez, Francois Fleuret, and Pascal Frossard. 2025. Lines: Post-training layer scaling prevents forgetting and enhances model merging. *Preprint*, arXiv:2410.17146.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *arXiv preprint*. Version Number: 2.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. *Preprint*, arXiv:2401.00368.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024a. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *Preprint*, arXiv:2412.13663.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024b. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *Preprint*, arXiv:2412.13663.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oğuz. 2021. Answering complex open-domain questions with multi-hop dense retrieval. *Preprint*, arXiv:2009.12756.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Retrieval meets long context large language models. *Preprint*, arXiv:2310.03025.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, and 9 others. 2025. Qwen2.5-1m technical report. *Preprint*, arXiv:2501.15383.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, and 1 others. 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.

Zijie Zhong, Hanwen Liu, Xiaoya Cui, Xiaofan Zhang, and Zengchang Qin. 2025a. Mix-of-granularity: Optimize the chunking granularity for retrieval-augmented generation. *Preprint*, arXiv:2406.00456.

Zijie Zhong, Hanwen Liu, Xiaoya Cui, Xiaofan Zhang, and Zengchang Qin. 2025b. Mix-of-granularity: Optimize the chunking granularity for retrieval-augmented generation. *Preprint*, arXiv:2406.00456.

Yang Zhou, Hongyi Liu, Zhuoming Chen, Yuandong Tian, and Beidi Chen. 2025. Gsm-infinite: How do your llms behave over infinitely increasing context length and reasoning complexity? *Preprint*, arXiv:2502.05252.

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

A *ConTEB* Details

This appendix describes the data generation process employed in this project. The methodology varies based on the dataset source, but generally, long documents are segmented into smaller chunks. If preexisting queries are available, they are mapped to relevant chunks using either provided answer spans (e.g., SQuAD) or tagged using GPT-4o. In cases where queries are unavailable, a large language model (LLM) generates them before associating them with the relevant text segments. This approach, illustrated in 5, is systematically applied across multiple datasets.

A.1 Wiki-based Datasets

Football and Geography are our two wiki-based datasets, focusing on the Sports and Geography domains.

Wikipedia Data Extraction The pipeline first retrieves Wikipedia summaries for a given person using the `wikipediaapi` library. The extracted summary is then split into paragraphs.

Text Rephrasing Each paragraph from the Wikipedia summary undergoes a rephrasing process to remove direct mentions of the person's name while maintaining the original context. The rephrased text replaces names with pronouns such as 'he' or 'she'. This transformation is performed using the GPT-4o model via the following prompt:

> *Here is a Wikipedia article:* `[Full Wikipedia Summary]` *Can you rephrase the following paragraph to remove all mention of the name of the person the article is about? You can leave other names as is and can replace the name with words such as 'he/she' or other generic paraphrases.* `[Paragraph to be rephrased]`

Question Generation For each paragraph in the summary, the model generates three questions re-

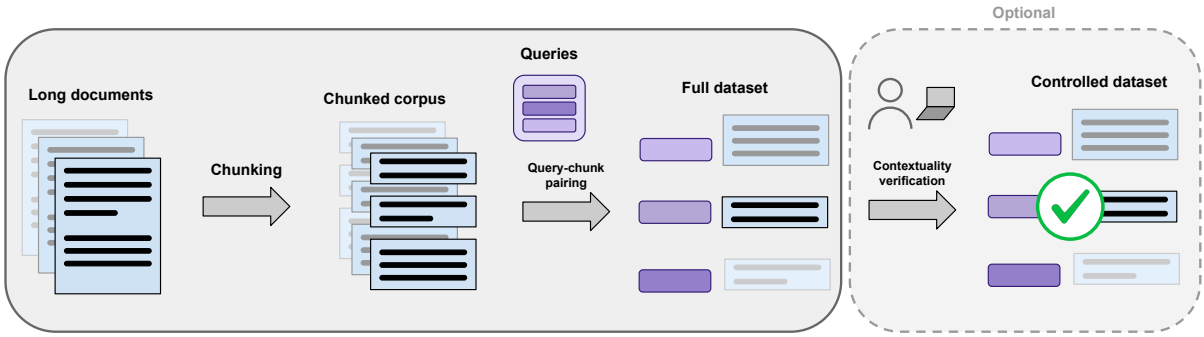

Figure 5: Benchmark creation process.

	# Queries	# Docs	Tokens/Chunk	Chunks/Doc	Query generation	Chunk Annotation
In Domain						
Chunked MLDR	100	100	170.5	15.4	Synthetic	Synthetic
NarrativeQA	8575	355	154.5	4.9	Human	Synthetic
SQuAD Chunked	2067	2067	19.1	8.5	Human	Manual
Out-Of-Domain						
Football	1767	224	86.1	3.6	Synthetic	Human in the loop
Geography	5283	530	113.6	4.3	Synthetic	Human in the loop
Insurance	120	1	80.7	60	Human	Manual
Covid-QA	1111	115	153.9	29.1	Human	Synthetic
ESG Reports	36	30	205.5	123.4	Human	Manual
NanoBEIR*	650	4363.3	199.4	1	Human	Manual

Table 3: Merged *ConTEB* dataset details and statistics. Controlled datasets are highlighted in **bold blue**. Values marked − indicate missing statistics in the source tables. [MF: **update this and add ESG stats**] [MC: **add context type?**]

lated to the person. The questions explicitly mention the person's name but do not include other named entities such as dates or proper nouns. The generation follows this structured prompt:

> *Here is a Wikipedia article:*
>
> `[Full Wikipedia Summary]`
>
> Using specifically the following paragraph, can you ask 3 questions related to the person the article is about? Each question must mention the name of the person, but the question should not contain other named entities (dates, other proper nouns). Format the response as a Python list of strings and do not output anything else.
>
> `[Paragraph to be used for question generation]`

A.2 NarrativeQA, COVID-QA, MLDR

NarrativeQA (literature), *MLDR* (encyclopedic) and *Covid-QA* (medical) consist of long documents, associated to existing sets of question-answer pairs.

We chunk these documents, and use GPT-4o to annotate which chunk, among the gold document, best contains information needed to answer the query. Since chunking is done *a posteriori* without considering the questions, chunks are not always self-contained and eliciting document-wide context can help build meaningful representations.

Synthetic Query Generation: To extend MLDR for our training dataset, OpenAI's GPT-4o model is prompted to generate 20-50 realistic queries per document, ensuring that each query aligns with the content of at least one chunk. This is on top of the queries that are already incuded in the dataset. Synthetic queries are included only in our training dataset.

A.3 Insurance

Insurance is composed of a long document with insurance-related statistics for each country of the European Union. Countries are often not referred to in-text, but only once in the section title. Therefore, certain chunks require knowledge of their position within the document to be properly disambiguated from others. Questions are manually

crafted to require structural understanding for accurate chunk matching. This process, in addition to manual verification of the contextuality quality, makes *Insurance* a controlled dataset. Since questions are crafted after the chunking process, the annotation results directly from the manual question generation process.

A.4 SQuAD

SQuAD is an extractive QA dataset with questions associated to passages and annotated answer spans, that allow us to chunk individual passages into shorter sequences while preserving the original annotation.

A.5 ESG Reports

ESG Reports contains long documents from the fast-food industry, with manually annotated query-page pairs from the ViDoRe Benchmark 2, originally thought for visual retrieving[7]. We convert all documents to text, chunk them, and re-annotate the resulting passages by hand, filtering out queries that relied solely on visual aspects (e.g., tables, graphs).

A.6 Training Data Statistics

Table 4 displays information about the training data. Our refined version of MLDR forms a large part of the training corpus. We can see that the majority of chunks are used as positives at least once, ensuring that the model is not biased towards the position of the chunk in the sequence.

	MLDR	NarrativeQA	SQuAD	Total
Number of Docs	8467	972	442	9881
Number of Chunks	213001	5219	14367	232587
Number of Queries	211933	27953	67355	307241
Number of Chunks per Doc	25.2	5.4	32.5	23.5
% Chunks with associated Query	94.6%	81.9%	100.0%	94.61%
Number of Tokens per Doc	3962.6	819.1	4966.1	3698.2
Number of Tokens per Query	16.7	21.9	12.5	16.3

Table 4: Training Dataset Statistics

B Implementation Details

B.1 Sequence prefixes

ModernBERT-based models are trained with query and document prefixes. We apply the same approach in our training and inference frameworks. After several tests, we opt for using a single document prefix for the Late Chunking sequence, instead of adding a document prefix at the beginning of each chunk inside the same sequence. We separate chunks with [SEP] tokens to let the model understand the concept of chunks during its token embedding computation.

B.2 Late Interaction Models

We leverage the `pylate`[8] library for the Late Interaction implementation. For training LI models with InSeNT, we adapt the LI mechanisms to incorporate it with Late Chunking in our own codebase. In particular, we do not use token skiplists at inference time, and use a single document prefix for the whole document sequence.

C Additional Results

C.1 Training with concatenated short documents

Results of training an InSeNT model with concatenated short document data (using the Nomic dataset) are available in Table 5. Short docs are clustered from the `nomic-supervised` dataset (Nussbaum et al., 2024) following Morris and Rush (2024). This approach did not yield promising results, proving that natively long documents are necessary to induce relevant in-sequence signal.

C.2 Full ablation results on λ_{seq}

We show the results of the different values for λ_{seq} on all our evaluation sets.

C.3 Extending context beyond 8192 tokens

ModernBERT was trained on documents of up to 8192 tokens (Warner et al., 2024a). Its Late Interaction counterpart, GTE-ModernColBERT, was exclusively fine-tuned on documents of no more than 300 tokens. However, its generalization capabilities to longer documents have been shown by its developers (Chaffin, 2025a), hinting at the fact that further research along those lines could be tried for both the bi-encoder and the LI variants.

Based on these results, we tried two approaches to handle documents longer than 8192 tokens with ModernBERT (necessary for the ESG reports dataset): computing Late Chunking with a context of max. 8192 tokens in an sliding window fashion (computing chunk embeddings in several forward passes of 8192 tokens, with 10 overlapping chunks between the various windows), and naively feeding the complete documents to the embedder.

[7] https://huggingface.co/datasets/vidore/restaurant_esg_reports_beir

[8] https://github.com/lightonai/pylate

	MLDR	SQuAD	NarrativeQA	Football	Geography	COVID-QA	Insurance	NanoBEIR	Average	Runtime (s)
MB	78.4	73.4	77.9	19.1	56.2	**61.7**	12.4	**63.2**	55.3	40.0
MB+InSeNT(Nomic)	77.8	76.0	76.2	26.2	62.7	38.8	63.7	59.9	60.2	36.3
MB+Late Chunking	78.5	77.1	75.8	54.6	89.6	40.0	41.0	**63.2**	65.0	36.3
Ours: MB+InSeNT	**88.7**	**80.9**	**81.3**	**63.9**	**90.7**	56.0	**100.0**	60.4	**77.8**	36.3

Table 5: Evaluation (nDCG@10) of baseline models and our proposed method on *ConTEB*. We show MB+InSeNT(Nomic) behaves like a non-contextual model after training on independant documents concatenated in a single sequence.

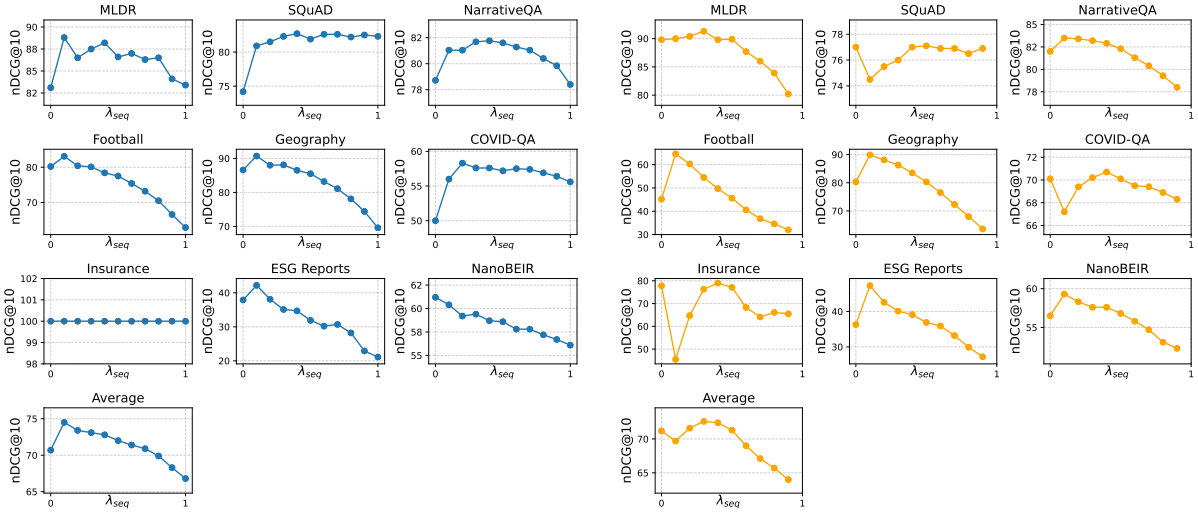

Figure 6: Evaluation results for varying λ_{seq} values. Left: ModernBERT-Large. Right: GTE-ModernColBERT. Trends vary across the datasets depending on their nature.

To our surprise, the latter worked better by a large margin (43.1 on ESG as reported in 2, vs 25.4 for the sliding window approach), so we reported the results of this approach. Further studies could be led to better understand the dynamics underlying this extension.