LMK > CLS
Landmark Pooling Improves Long-Context Dense Embeddings

Anonymous ACL submission

Abstract

Representation learning is central to tasks such as search, clustering, classification, and reranking. State-of-the-art sequence encoders compress variable-length token sequences into a single vector via pooling, most commonly using a special `[CLS]` token or mean pooling over token embeddings. We identify systematic limitations of these strategies: `[CLS]` pooling biases representations toward early tokens and can under-represent distributed evidence, while mean pooling dilutes salient local signals, often degrading short-context performance. To address this, we introduce *Landmark* (`LMK`) pooling, which partitions sequences into chunks, inserts landmark tokens between chunks, and forms the final representation by mean-pooling landmark embeddings. `LMK` improves long-context extrapolation while retaining strong short-context performance with only a small number of additional special tokens. Empirically, `LMK` matches or outperforms existing pooling methods on short-context retrieval benchmarks and yields substantial gains on long-context tasks, including +7 on MLDR and ∼+3 on COIR when trained on MS MARCO (Table 1). When trained on multilingual data, it further improves MLDR (13 languages) by +13.8, MultiEURLEX (23 languages) by +5.7, and LongEmbed by +23.3 (Table 2). These results show that `LMK` provides a practical alternative to existing pooling strategies, particularly for long-context extrapolation.

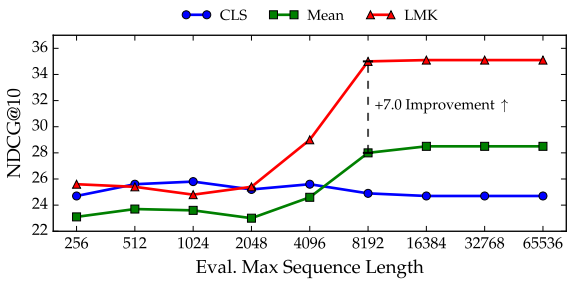


Figure 1: MLDR (English) dataset long context retrieval performance for different pooling strategies using *modernbert-base* finetuned on MSMarco Passages.

1 Introduction

Traditionally, document-level representations were constructed using vector space based feature extraction methods such as CBOW and Skip gram, which aggregate word-level statistics or embeddings into fixed size representations (Mikolov et al., 2013; Clinchant and Perronnin, 2013; Zuccon et al., 2015). These approaches were later supplanted by neural sequence encoders, including LSTMs and Transformers (Vaswani et al., 2017; Peters et al., 2018), primarily because classical methods fail to preserve rich document-level semantics and rely on fixed feature heuristics that cannot compete with contextualized representations learned by neural models (Li et al., 2020).

A common application of neural text encoders is constructing fixed-dimensional sentence representations for downstream tasks such as retrieval, classification, and clustering (Li et al., 2020; Yan et al., 2021; Feng et al., 2022). Formally, given an input sequence $\mathcal{X}$, a tokenizer $\mathcal{F}_{\text{tok}}$ produces a token sequence $\mathcal{X}_{\text{tok}} = (x_1, x_2, \ldots, x_t)$, which is augmented with special tokens such as `[CLS]` and `[SEP]`. The resulting sequence is passed through an encoder θ_{enc} to obtain contextualized representations $\mathbf{H}^{\text{enc}} = (\mathbf{h}_{\texttt{[CLS]}}, \mathbf{h}_1, \ldots, \mathbf{h}_t, \mathbf{h}_{\texttt{[SEP]}})$, where each vector corresponds to a token in the sequence.

A key challenge is converting the variable-length token representations $\mathbf{H}^{\text{enc}}$ into a single fixed-dimensional representation of the input sequence, denoted $\mathcal{X}^{\text{enc}}$. This is typically achieved using a pooling function $\mathcal{P}$. Existing work most commonly employs simple strategies such as `[CLS]` pooling,

$$\mathcal{P}_{\text{CLS}}(\mathbf{H}^{\text{enc}}) = \mathbf{H}^{\text{enc}}[0, :] = \mathbf{h}_{\texttt{[CLS]}} \qquad (1)$$

or mean pooling,

$$\mathcal{P}_{\text{MEAN}}(\mathbf{H}^{\text{enc}}) = \frac{1}{|\mathbf{H}^{\text{enc}}|} \sum_{i=0}^{|\mathbf{H}^{\text{enc}}|} \mathbf{h}_i. \quad (2)$$

Intuitively, mean pooling provides a holistic aggregation by uniformly combining all token representations; however, this averaging can dilute salient global features. In contrast, [CLS] pooling relies on a single special token to encode all relevant information into a fixed-dimensional vector. While conceptually appealing, this imposes a strong representational bottleneck on a single position and, as we observe, often fails to generalize effectively to long contexts, an increasingly important property for modern text encoders.

To address these limitations, we investigate a simple yet effective alternative termed *Landmark* (LMK) pooling. Our approach is inspired by Luo et al. (2024), who introduced landmark tokens to mitigate abrupt chunking effects in retrieval-augmented generation systems. We extend this idea to Dense Passage Retrieval (DPR) by inserting multiple special *landmark* tokens at regular intervals between fixed-size chunks of input tokens, rather than relying on a single token such as [CLS]. The final sequence representation is then obtained by mean-pooling only the landmark token embeddings. We posit that this strategy alleviates the representational bottleneck of [CLS] in long-context settings while preserving salient global features often diluted by standard mean pooling. We provide empirical evidence supporting these claims and show that LMK pooling offers a simple and effective mechanism for improving long-context generalization in neural text encoders, making it a practical alternative to existing pooling strategies.

Our contributions: (1) We introduce LMK pooling, a simple alternative to common pooling methods that **delivers substantial gains on long-context embedding benchmarks** across domains and languages. **(2)** Through systematic evaluation, we identify inherent limitations of popular pooling mechanisms for long-context generalization and characterize their strengths and weaknesses. **(3)** We demonstrate that LMK pooling is robust across training regimes and compatible with existing pre-training and retrieval optimizations.

2 Understanding Pooling Methodologies

In this section, we analyze existing pooling methodologies as we motivate the need for LMK pooling.

2.1 CLS pooling

[CLS] pooling is a widely used strategy in text embedding models. Most state-of-the-art approaches pretrain models with long-context masked language modeling (MLM) objectives and then fine-tune them for embedding tasks using [CLS] pooling on predominantly short-sequence datasets. We show that [CLS] pooling inherently biases representations toward early positions in the text, and that training on shorter sequences further amplifies this effect. To understand the origin of this behavior, we analyze how positional information is encoded in text encoders.

In Transformers, to generate contextualized sequence representations, each input passes through an attention block. Let the input sequence be $\mathcal{X} = [x_1, \ldots, x_t]$, with $x_i \in \mathbb{R}^d$. For a single attention head, we compute query, key, and value vectors:

$$q_i = W_Q x_i, \quad k_i = W_K x_i, \quad v_i = W_V x_i,$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_h}$ and d_h is head dimension. Attention scores, obtained from queries and keys, are used to weight the values to form contextualized representations.

Rotary Positional Embeddings (RoPE) (Su et al., 2024) encode positional information by applying a position-dependent rotation $R_{\theta,i}$ to the query and key vectors. The angular frequencies are defined as $\theta_j = b^{-\frac{2j}{d_h}}$, for $j = 0, \ldots, \frac{d_h}{2} - 1$, with b as the base frequency (typically set to 10,000). Defining $\tilde{q}_m = R_{\theta,m} q_m$ and $\tilde{k}_n = R_{\theta,n} k_n$, the attention logit between positions m and n is:

$$\begin{aligned} \text{Attn}(m,n) &:= \tilde{q}_m^\top \tilde{k}_n \\ &= q_m^\top R_{\theta,m}^\top R_{\theta,n} k_n \Rightarrow q_m^\top R_{\theta,n-m} k_n \end{aligned}$$

which shows that RoPE induces an explicit dependence on relative position, $n - m$.

Long-term decay for CLS: The rotation matrices in RoPE induce a relative distance-based attenuation of the attention logit $\tilde{q}_m^\top \tilde{k}_n$ for large relative offsets $|n - m|$, also referred to as the *long-term decay* of RoPE. Specifically, for a [CLS] token at position $m = 0$:

$$\tilde{q}_{\text{CLS}}^\top \tilde{k}_n = q_{\text{CLS}}^\top R_{\theta,n} k_n,$$

whose magnitude decreases as n increases. After softmax, this produces smaller attention weights for distant tokens, biasing the [CLS] representation toward early positions. The effect becomes

more pronounced for longer sequences, limiting the effectiveness of [CLS] pooling for long-context extrapolation. Increasing the RoPE base frequency (b) slows this positional decay and can improve long-context extrapolation at inference time (Zhang et al., 2024; Peng et al., 2024), but does not fully mitigate the underlying bias. Although Barbero et al. (2025) show that this decay vanishes when query and key vectors are assumed to follow a Gaussian distribution, this assumption does not hold in practice, as we demonstrate below.

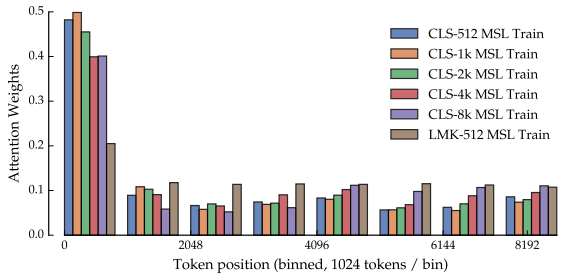


Figure 2: Normalized attention distribution over long MLDR documents (> 8192 tokens) for *mmBERT-base* fine-tuned with CLS and LMK pooling with varying maximum sequence lengths (MSL).

We provide empirical evidence of this bias in [CLS] pooling in Fig. 2. By analyzing final-layer attention weights assigned to pooling tokens ([CLS] and LMK) over long documents from the MLDR English subset, we observe a clear concentration of attention toward early sequence positions. Fine-tuning on longer sequences reduces this bias but does not eliminate it, and the effect worsens when extrapolating beyond the training sequence length. In contrast, LMK pooling exhibits lower bias despite being trained on only 512 tokens.

2.2 Mean and Latent Attention pooling

Given token embeddings $\mathbf{H}^{\text{enc}} \in \mathbb{R}^{S \times D}$, mean pooling (defined in Eq. (2)) averages all token embeddings across each latent dimension. Unlike [CLS] pooling, which relies on positional interactions, mean pooling is position-independent. While simple and effective, it causes all tokens to contribute uniformly during aggregation, potentially diluting salient global semantic signals. Moreover, mean pooling assigns equal weights to all tokens regardless of their semantic importance, thus tokens carrying little task-relevant information also contribute to final embeddings in typical ℓ_2-normalized (norm bounded) encoder representations.

Consequently, mean pooling can underperform approaches such as [CLS] pooling in short-context settings, where the model learns an explicit aggregation mechanism through attention. At the same time, we observe that mean pooling extrapolates well to long contexts at inference time and faces no inherent RoPE-related limitations in doing so. However, despite this strong extrapolation behavior, it can still underperform on tasks requiring robust global semantic representations.

Lee et al. (2025) introduce *Latent Attention* to address the limitation of assigning equal importance to all tokens. The method uses learnable latent vectors to selectively attend to different token embeddings. Formally, let $\mathbf{H}^{\text{enc}} = (\mathbf{h}_{\texttt{[CLS]}}, \mathbf{h}_1, \ldots, \mathbf{h}_t, \mathbf{h}_{\texttt{[SEP]}})$ denote the sequence of embeddings produced by the encoder, where each $\mathbf{h}_i \in \mathbb{R}^d$. Let $\mathbf{L} \in \mathbb{R}^{\ell \times d_l}$ denote a set of learnable latent vectors, where ℓ is the number of latents and d_l is the latent head dimension. A latent attention module first computes the query, key, and value representations for $\mathbf{H}^{\text{enc}}$ and $\mathbf{L}$ as

$$q_i = W_Q \mathbf{h}_i, \quad k = W_K \mathbf{L}, \quad v = W_V \mathbf{L},$$

where $W_Q \in \mathbb{R}^{d \times d_h}$ and $W_K, W_V \in \mathbb{R}^{(\ell \times d_l) \times d_h}$, and d_h denotes the head dimension corresponding to $\mathbf{h}_i$.

$$\begin{aligned} \mathcal{Y}_i &= \text{Cross-Attn}(\text{Prenorm}(q_i), k, v) + q_i, \\ \mathcal{Z}_i &= \text{FFN}(\text{Prenorm}(\mathcal{Y}_i)) + \mathcal{Y}_i, \\ \mathcal{P}_{\text{LATENT-ATTN}} &= \frac{1}{|\mathcal{Z}|} \sum_{i=0}^{|\mathcal{Z}|} \mathcal{Z}_i. \end{aligned} \tag{3}$$

Here, Prenorm refers to standard LayerNorm applied prior to the transformation, and FFN denotes the standard feed-forward network used in Transformer architectures.

This approach introduces additional parameters, including learnable latent vectors and feed-forward layers, which are used to determine how the latents are reweighted based on the sequence representations to produce new embeddings. The representations generated from these latent vectors are then mean pooled to obtain a single sequence-level representation. Conceptually, this procedure is equivalent to mean pooling and, as such, should not introduce a bias toward long-context extrapolation, which is consistent with our empirical results in Section 5. We therefore posit that latent attention inherits the same limitations as mean pooling, while incurring additional parameter and computational overhead that yields limited empirical gains.

3 Landmark (LMK) Pooling

Algorithm 1 Landmark Tokenization

Require: Text sequence $\mathcal{X}$, Tokenizer $\mathcal{F}_{\text{tok}}$, Granularity g, Landmark token t_{LMK}
Ensure: Token IDs $\mathcal{X}_{\text{tok}}$, Attention mask $\mathcal{X}_{\text{mask}}$
$T \leftarrow \mathcal{F}_{\text{tok}}(\mathcal{X})$ ▷ Tokenize (no special tokens)
$Z \leftarrow \text{Split}(T, g)$ ▷ Split token sequence
$\mathcal{X}_{\text{tok}} \leftarrow [t_{\text{CLS}}] \oplus z_1 \oplus [t_{\text{LMK}}] \oplus z_2 \cdots \oplus [t_{\text{LMK}}] \oplus z_n \oplus [t_{\text{LMK}}]$
$\mathcal{X}_{\text{mask}} \leftarrow \mathbb{1}(\mathcal{X}_{\text{tok}} \neq t_{\text{PAD}})$
return $\mathcal{X}_{\text{tok}}, \mathcal{X}_{\text{mask}}$

Algorithm 2 Landmark (LMK) Pooling

Require: Token embeddings $\mathbf{H}^{\text{enc}} \in \mathbb{R}^{S \times D}$, Token IDs $\mathcal{X}_{\text{tok}}$, Attention mask $\mathcal{X}_{\text{mask}}$, Landmark token t_{LMK}
Ensure: Sentence embedding $\mathcal{X}^{\text{enc}} \in \mathbb{R}^{D}$
$\mathcal{I}_L \leftarrow \{\, i \mid \mathcal{X}_{\text{tok},i} = t_{\text{LMK}} \ \wedge \ \mathcal{X}_{\text{mask},i} = 1 \,\}$ ▷ LMK index set
$\mathcal{X}^{\text{enc}} \leftarrow \frac{1}{|\mathcal{I}_L|} \sum_{i \in \mathcal{I}_L} \mathbf{H}^{\text{enc}}_{i,:}$ ▷ Mean over LMK
return $\mathcal{X}^{\text{enc}}$

We highlight the key limitations of these pooling approaches, while noting the benefit that they provide. [CLS] pooling biases attention towards nearer tokens, attenuating the attention from distant positions. However, the [CLS] token acts as a learned selector that adaptively attends to different parts of the sequence- a desirable feature in representation learning. Furthermore, mean pooling dilutes salient features by uniformly averaging over all tokens, yet it allows multiple tokens to independently asses the full sequence, which may be more effective than single-token [CLS] representation. Finally, latent attention relies on learnt latent vectors to reweigh sequence embeddings, thereby adding undesirable complexity at inference time.

Landmark (LMK) pooling provides a simple yet effective solution to the limitations discussed above. It introduces two key modifications to standard text encoding: (1) a change to the tokenization procedure and (2) a pooling strategy that depends on the positions of LMK tokens. Instead of relying on a single special token, LMK pooling inserts multiple special tokens throughout the input sequence. **We use the SEP token as the LMK token, as it already serves as a natural delimiter in most language encoders**[1].

We explore several strategies for placing these tokens along the sequence. **Sentence Chunking** involves splitting the text into sentence prior to tokenization and inserting LMK tokens after each sentence. However, this approach may not generalize well to other domains, such as code or certain natural languages, where clear sentence delimiters may be absent. Alternatively, **fixed length chunking** inserts LMK tokens at fixed intervals after tokenization, but this can introduce positional bias while making performance sensitive to the granularity chosen during fine-tuning. To address these issues, **variable chunking** randomly samples the LMK insertion interval from a fixed set of predefined intervals. This encourages robustness to granularity choice and allows a single model to support different granularities at inference depending on the task. The final sequence representation is obtained by mean pooling the embeddings of all LMK tokens.

While inserting additional special tokens increases the effective sequence length and incurs extra computation, the overhead is modest in practice since LMK tokens are added at coarse intervals. For example, on the LongEmbed benchmark with a maximum evaluation sequence length (MSL) of 32K tokens, inserting an LMK token every 128 tokens adds only 256 additional tokens to the sequence while significantly outperforming alternative pooling mechanisms as seen in Table 1. Formal pseudocode for the tokenization procedure is provided in Algorithm 1, and the pooling operation is detailed in Algorithm 2.

4 Experimental Setup

In this section, we describe the experimental setup used to obtain our results. First, we train models on standard retrieval datasets using different pooling strategies and compare their performance on widely used retrieval benchmarks. Second, we analyze the effect of long-context evaluation across different domains. Third, we study how long-context training influences the extrapolation behavior of pooling mechanisms. We divide our experiments into **English** and **Multilingual** settings, and further into short- and long-context training.

English: We select widely used Transformer-based text encoders that support long-context training and fine-tuning, namely *gte-en-mlm-base* and *ModernBERT-base* (Warner et al., 2025). Standard retrieval training datasets, including MS MARCO passage and document ranking as provided by Li et al. (2025a), are used. Training includes hard negatives and distillation scores, with 7 hard negatives for passage ranking and 1 for document ranking. Models are trained for 5k steps with a learning

[1] For tokenizers that do not have a SEP token, the EOS token is an appropriate alternative.

rate of 2×10^{-5}. Passage ranking uses an effective query batch size of 2,048 with a maximum sequence length of 512 tokens, while document ranking uses a batch size of 256 with sequences up to 8,192 tokens.

To mitigate bias from domain overlap, we evaluate on a diverse set of out of domain benchmarks. In domain performance is reported on the MS MARCO Dev set, followed by short context evaluations on BEIR (Thakur et al., 2021), MTEB-v2 (Enevoldsen et al., 2025), and MIRACL Retrieval (Zhang et al., 2023). For long context evaluation, we use MLDR, the COIR code retrieval benchmark (Li et al., 2025b), and LongEmbed, which includes documents of diverse lengths, often exceeding the 8k token pretraining context, with dispersed target information (Zhu et al., 2024).

Multilingual: For multilingual experiments, we use the multilingual counterpart of ModernBERT, *mmBERT-base*. Models are fine-tuned on the multilingual training mixture from Chen et al. (2024), which includes diverse datasets with hard negatives and multilingual re-ranker distillation scores. Training is performed for 10k steps with a learning rate of 2×10^{-5}, an effective query batch size of 1024, maximum sequence length of 512 tokens, and 7 hard negatives. In ablations, we additionally fine-tune *mmBERT-base* and *gte-multilingual-base* with sequence lengths up to 8,192, reducing the batch size to 64. We evaluate multilingual retrieval on short-context benchmarks such as MIRACL Hard Negatives (18 languages) and long-context benchmarks including MLDR (13 languages) and LongEmbed. We additionally evaluate zero-shot multilingual long-document classification on MultiEURLEX across 23 languages (Chalkidis et al., 2021), reporting Macro-F1 scores.

Pooling Strategies: We compare LMK pooling against several baselines from prior work, including CLS, Mean, and Latent Attention (Attn.) pooling with 64 latents, described in Section 2.2. We also evaluate **MultiCLS**, where a model trained with CLS pooling is modified at inference by inserting CLS tokens at fixed intervals and mean-pooling their representations, analogous to inference-time LMK but without LMK-style training. As shown by Chen et al. (2024), this improves MLDR performance for CLS without long-context training. Finally, we include a **Mean@k** baseline, which computes the final representation by mean-pooling every k-th token, to assess whether special pooling tokens are necessary for strong performance.

For Landmark pooling, we compare different LMK insertion granularities as described in Section 3. We consider: (1) a sentence-based splitter using a heuristic English sentence splitter to study the role of sentence boundaries, (2) a fixed splitter that inserts LMK tokens at regular intervals after tokenization, and (3) a variable splitter where the granularity is uniformly sampled from $\{32, 64, 128, 256\}$ during training, enabling robustness and support for multiple granularities at inference. All pooling strategies are trained under identical hyperparameters and data settings to eliminate training-related biases. During evaluation, we use a maximum sequence length of 8,192 for all tasks except LongEmbed, which uses 32,768, and explicitly report pooling strategies and granularities in the corresponding tables and figures. **Importantly, LMK evaluations do not increase the token budget to account for additional special tokens**. We report standard retrieval metrics such as NDCG@10 and Precision@1. Additional training and evaluation details are provided in Appendix E, with dataset descriptions in Appendix F.

5 Results & Discussion

We aim to answer the following research questions: 1) How does LMK pooling compare with other pooling strategies across short- and long-context tasks, and out-of-domain settings? 2) Is the positional bias learned by CLS pooling a modeling limitation or a data artefact? 3) Can training with longer contexts mitigate this bias? 4) Is LMK pooling subject to positional bias, and how does it generalize across different downstream tasks? We also report additional experiments analyzing LMK's ability to capture local information, including its application to decoder-based embedding models (Appendices B and D).

5.1 LMK Pooling Performance

We examine English retrieval performance by traning on the MSMarco Passage dataset, showing results for *ModernBERT-base* and *gte-en-mlm-base* in Table 1 and Table 7 (Appendix) respectively. We observe that LMK pooling peforms marginally better than all pooling baselines on the in-domain MSMarco Dev set. For short-context tasks, LMK pooling maintains comparable performance to CLS pooling, both outperforming Mean and Latent attention pooling, indicating that LMK is a strong al-

Finetuning		Evaluation		In-Domain	Out-of-Domain (Short Context)				Out-of-Domain (Long Context)			
				MSMarco Dev	**BEIR (15)**	**MTEBv2 (10)**	**MIRACL Dev**	**Avg. Short**	**MLDR Test**	**COIR (8)**	**LongEmbed (6)**	**Avg. Long**
Pool	**Gran.**	**Pool**	**Gran.**	NDCG@10	NDCG@10	NDCG@10	NDCG@10		NDCG@10	NDCG@10	P@1 | NDCG@10	
CLS	-	CLS	-	39.8	42.8	44.2	48.7	45.2	24.9	43.5	43.3	37.2
CLS	-	MultiCLS	128	39.0	33.6	31.2	31.0	31.9	5.1	35.4	43.9	28.1
Mean	-	Mean	-	39.8	41.6	43.3	48.0	44.3	28.0	44.1	62.7	44.9
Latent Attn	-	Latent Attn	-	40.1	41.3	43.1	48.3	44.2	24.5	43.3	62.8	43.5
Mean@k	64	Mean@k	64	38.6	39.7	41.4	44.1	41.7	15.7	39.8	44.8	33.4
LMK	32	LMK	32	39.7	43.9	45.9	48.5	46.1	34.8	46.4	60.2	47.1
	64		64	39.9	43.5	45.1	48.2	45.6	33.6	46.6	60.0	46.7
	128		128	40.0	42.9	44.0	48.4	45.1	30.8	46.3	54.1	43.7
	256		256	39.9	41.8	42.8	47.0	43.9	34.1	46.4	58.2	46.2
LMK	Variable	LMK	32	40.3	44.0	45.8	48.6	46.1	35.0	47.0	62.4	48.1
			64	40.2	43.8	45.5	48.6	46.0	34.7	46.9	62.3	48.0
			128	40.2	43.6	45.1	48.1	45.6	32.1	46.8	59.0	46.0
			256	40.2	43.1	44.5	48.1	45.2	28.7	46.4	55.5	43.5
LMK	Sentence	LMK	Sentence	40.2	43.6	45.4	48.2	45.7	28.0	42.7	62.6	44.4

Table 1: Comparison of pooling strategies across in-domain and out-of-domain retrieval benchmarks for *ModernBERT-base* fine-tuned on MSMarco passage data. We report the pooling strategy and granularity (gran.) used during both fine-tuning and evaluation.

Pool	FT.	Eval.	In-Domain		Out-of-Domain	
	Gran.	**Gran.**	**MIRACL (18)**	**MLDR (13)**	**M-Eurlex (23)**	**LongEmbed (6)**
			NDCG@10	NDCG@10	F1	P@1 | NDCG@10
CLS	–	–	57.6	24.9	27.5	44.1
Mean	–	–	56.1	17.0	27.5	48.4
Latent	–	–	56.6	19.6	28.0	46.9
LMK	128	128	**58.0**	37.2	31.5	62.1
LMK	Variable	32	57.8	**38.7**	**33.7**	**70.7**
		64	57.9	38.6	33.4	70.7
		128	57.9	38.3	33.2	70.2
		256	57.7	38.0	33.0	69.8

Table 2: Comparison of pooling strategies on in-domain and out-of-domain retrieval benchmarks for *mmBERT-base* fine-tuned on multilingual BGE-m3 data.

ternative to CLS for such tasks. For long-context tasks, LMK pooling significantly outperforms all other baselines. Moreover, we find that using *Variable* splitting during LMK training generalizes well while providing the flexibility of choosing the pooling granularity at inference time.

To assess whether LMK pooling generalizes across languages, we report the performance of *mmBERT-base* on widely uses multilingual retrieval and long-context classification benchmarks after contrastive finetuning. Table 2 shows that LMK pooling maintains substantial improvements on long-context evaluations across in and out-of-domain multilingual tasks. Full results are provided in Appendix A.

5.2 Long-Context Performance: Data Artefacts or Modeling Limitations?

Having demonstrated the effectiveness of LMK pooling in long-context generalization, we evaluate whether this persists when any context-length bias of the training data is reduced by adding very long documents to the training mix. Specifically, we use the MSMarco Document Ranking dataset, increase the training context length to 8k, and report results for two models trained in this long-context setting in Table 3.

For *gte-en-mlm-base*, CLS pooling shows the strongest performance on short-context tasks, with LMK pooling yielding comparable results. In contrast, on long-context tasks, Mean pooling outperforms both CLS and LMK, with CLS substantially underperforming. For *ModernBERT-base*, LMK pooling is the most consistent method, performing well across both short- and long-context tasks. CLS achieves comparable scores on LongEmbed but primarily due to the Passkey Retrieval dataset; on other long-context benchmarks, CLS substantially underperforms (Table 9).

Although the MSMarco document ranking dataset is a popular long-context training source, its passages are not sufficiently long to enable robust long-context learning (Fig. 13). We instead finetune on the same multilingual training data employed in our earlier experiments, which includes the MLDR training set with significantly longer documents. All models are finetuned with a maximum sequence length of 8192 and a batch size of 64 for 5k steps. We vary the training sequence length for both CLS and LMK pooling using *mmBERT-base*, and report results in Table 4. As training sequence length increases, CLS and LMK achieve comparable performance on MLDR sug-

Model	Finetuning		Evaluation		In-Domain	Out-of-Domain (Short Context)				Out-of-Domain (Long Context)			
					MSMarco Dev	BEIR (15)	MTEBv2 (10)	MIRACL Dev	Avg. Short	MLDR Test	COIR (8)	LongEmbed (6)	Avg. Long
	Pool	Gran.	Pool	Gran.	NDCG@10	NDCG@10	NDCG@10	NDCG@10		NDCG@10	NDCG@10	P@1 | NDCG@10	
gte-en-mlm-base	CLS	-	CLS	-	28.7	**37.6**	**39.0**	32.3	**36.3**	29.1	30.1	49.1	36.1
	Mean	-	Mean	-	28.3	36.3	37.7	29.0	34.3	30.0	**32.5**	**61.6**	**41.4**
	LMK	Variable	LMK	32	28.9	37.1	38.1	32.1	35.8	**31.4**	30.9	59.8	40.7
ModernBERT-base	CLS	-	CLS	-	26.0	38.0	39.8	32.0	36.6	29.7	44.2	58.3	44.1
	Mean	-	Mean	-	26.1	37.1	38.4	31.4	35.6	31.3	42.3	**62.4**	**45.3**
	LMK	Variable	LMK	32	26.0	**38.7**	**40.4**	**33.1**	**37.4**	**32.2**	**45.6**	58.1	**45.3**

Table 3: Comparison of pooling strategies across in-domain and out-of-domain retrieval benchmarks for *gte-en-mlm-base* and *ModernBERT-base* when fine-tuned with MSMarco document data.

FT Train	In-Domain		Out-of-Domain			
	MLDR (13)		Multi-EURLEX (23)		LongEmbed (6)	
MSL	CLS	LMK	CLS	LMK	CLS	LMK
	NDCG@10		F1 Score		P@1 | NDCG@10	
512	31.7	**36.6**	30.5	**32.9**	46.1	**70.6**
1024	32.5	**40.9**	30.1	**33.1**	45.4	**71.1**
2048	39.8	**43.6**	30.0	**32.0**	52.2	**72.6**
4096	**46.3**	45.9	29.4	**31.7**	52.5	**73.3**
8192	49.1	**49.6**	30.2	**31.5**	56.2	**71.3**

Table 4: Effect of fine-tuning maximum sequence length (MSL) on long-context embedding tasks for *mmBERT-base*, comparing CLS and LMK pooling.

Pool	FT.	Eval.	In-Domain		Out-of-Domain	
	Gran.	Gran.	MIRACL (18)	MLDR (13)	M-Eurlex (23)	LongEmbed (6)
			NDCG@10	NDCG@10	F1	P@1 | NDCG@10
CLS	–	–	**66.7**	**58.7**	13.9	66.1
Mean	–	–	66.3	58.3	14.5	65.8
Latent	–	–	66.2	58.4	14.6	66.9
LMK	32	32	66.1	58.1	16.4	72.9
LMK	Variable	32	66.2	58.1	**18.0**	**73.1**

Table 5: Long-context finetuning ablations on *gte-multilingual-base* with a MSL of 8192.

gesting that CLS can partially mitigate long-context bias with longer training sequences. However, this does not generalize to out-of-domain datasets such as Multi-EURLEX and LongEmbed, where a substantial performance gap persists.

We conduct a similar experiment on *gte-multilingual-base*, (Table 5), and observe a consistent trend: while CLS, Mean, and Latent Attention pooling become competitive with LMK on MLDR, they substantially under-perform on out-of-domain long-context tasks. These results suggest indicate that long-context training can reduce but not eliminate the positional bias of these pooling methods.

5.3 Extension to Retrieval Pretraining

Retrieval oriented pretraining, as proposed in RetroMAE (Xiao et al., 2022; Xiao and Liu, 2022), is a popular self-supervised training method to learn effective text representations without the use of labeled data. It is often an intermediate pretraining step to strengthen vector embeddings before large-scale contrastive learning (Awasthy et al., 2025; Chen et al., 2024; Sturua et al., 2024). The method employs a two-tier architecture, where a standard encoder processes masked input sequences while a lightweight, bottleneck decoder attempts to reconstruct the original text from the encoder representations. While the CLS representation is typically used as an input to the enhanced decoder, we show that LMK pooling can be used instead, providing significant performance gains in downstream long context retrieval tasks.

Specifically, we follow the pretraining method proposed in Xiao et al. (2022) using about 225k samples from BookCorpus, Wikipedia and StackExchange, training for 2 epochs using a learning rate of 2×10^{-5} and a maximum sequence length of 8192 tokens. We use LMK pooling instead of the traditional CLS, experimenting with different granularities of inserting LMK tokens into the sequence, before mean pooling across these tokens to reconstruct the masked sentence with the decoder. Table 6 compares pretraining with LMK pooling and CLS pooling after finetuning on MS-Marco passage data. We observe consistent performance gains from LMK pretraining on *gte-en-mlm-base*, whereas for *ModernBERT-base*, both pooling strategies yield comparable results with no clear advantage.

6 Related Work

Training and Evaluation Datasets: Embedding training data have evolved from benchmarks such as MSMarco and NQ (Nguyen et al., 2016; Kwiatkowski et al., 2019) to large heterogeneous mixtures used to train modern text embedders (Chen et al., 2024; Li et al., 2025a). Evaluation has similarly expanded to benchmarks spanning mul-

Model	Finetuning Pool	Finetuning Gran.	Evaluation Pool	Evaluation Gran.	In-Domain MSMarco Dev	Out-of-Domain (Short Context) BEIR (15)	MTEBv2 (10)	MIRACL-HN EN	Avg. Short	Out-of-Domain (Long Context) MLDR EN	COIR (8)	LongEmbed (6)	Avg. Long
					NDCG@10	NDCG@10	NDCG@10	NDCG@10		NDCG@10	NDCG@10	P@1 | NDCG@10	
gte-en-mlm-base	CLS	–	CLS	–	42.3	44.0	44.7	47.9	45.5	27.3	36.6	52.0	38.6
	+ CLS RetroMAE pretraining				42.5	**44.7**	**45.5**	47.4	45.9	31.2	42.1	60.8	44.7
	LMK	Variable	LMK	256	**43.0**	44.2	44.8	48.1	45.7	31.7	37.6	65.2	44.8
	+ LMK RetroMAE pretraining				42.6	44.5	45.4	**48.4**	**46.1**	**36.6**	**43.3**	**67.7**	**49.2**
ModernBERT-base	CLS	–	CLS	–	39.8	42.8	44.2	**48.7**	45.2	24.9	43.5	43.3	37.2
	+ CLS RetroMAE pretraining				39.8	42.8	44.4	47.9	45.0	24.9	44.3	42.1	37.1
	LMK	Variable	LMK	32	**40.3**	**44.0**	**45.8**	48.6	**46.1**	**35.0**	**47.0**	62.4	**48.1**
	+ LMK RetroMAE pretraining				40.2	41.3	42.5	47.8	43.9	31.8	46.2	**66.4**	**48.1**

Table 6: Effect of RetroMAE pretraining followed by fine-tuning for CLS and LMK pooling across *gte-en-mlm-base* and *ModernBERT-base* models.

tiple domains, tasks, and languages (Thakur et al., 2021; Enevoldsen et al., 2025). However, **widely used benchmarks such as BEIR and MTEBv2 primarily focus on short sequences** (typically $\leq$512 tokens; see Appendix C). In contrast, many applications require robust long-context embedding evaluation, which remains underexplored, particularly regarding how representation and pooling choices affect performance (Li et al., 2025b; Zhu et al., 2024; Chalkidis et al., 2021).

Dense Embedders: Early work on dense embedding representations (Gao et al., 2011; Huang et al., 2013) learned token transformations by minimizing distance-based objectives between relevant queries and documents. These approaches were later superseded by stronger encoder architectures (Devlin et al., 2019; Warner et al., 2025) and improved training objectives (van den Oord et al., 2018), significantly improving representation quality. However, most methods still compress token representations into a single vector using simple pooling strategies, an aspect that remains underexplored for long-context retrieval.

Pooling Methodologies: There is limited consensus in dense passage retrieval on how to aggregate token representations into a single document embedding. Many recent models rely on CLS pooling (Awasthy et al., 2025; Zhang et al., 2024; Merrick et al., 2024), while others achieve comparable performance with mean pooling (Wang et al., 2022; Nussbaum et al., 2024). Some adopt learned weighting or hybrid strategies (Chen et al., 2024; Lee et al., 2025), with Chen et al. (2024) introducing multi-CLS pooling by inserting multiple CLS tokens at inference. Since these representations remain fixed in size regardless of document length, they limit the amount of preserved information (Santhanam et al., 2022; Weller et al., 2025), motivating more robust pooling strategies.

For long-context retrieval and retrieval-augmented generation, Luo et al. (2024) introduce Landmark tokens to capture chunk-level representations in autoregressive language models, but do not study their use for pooling long documents into bidirectional dense embeddings. In contrast, our work systematically analyzes pooling mechanisms across short- and long-context retrieval benchmarks. Additional studies have also highlighted limitations of existing pooling strategies; we refer readers to (Li et al., 2020; Chen et al., 2023; Lee et al., 2025; Ennadir et al., 2025) for further discussion.

7 Conclusion

In this work, we first highlight biases and shortcomings of existing pooling mechanisms across various embedding tasks. We then introduce Landmark (LMK) pooling, a simple yet effective method that partitions sequences into chunks, inserts landmark tokens, and mean-pools their embeddings. Extensive experiments across multiple encoder architectures, languages, and training setups show that LMK pooling yields substantial improvements on long-context evaluations while maintaining competitive performance on short-context retrieval, and enables robust extrapolation beyond training sequence lengths with minimal computational overhead. Further analysis of LMK embeddings shows that they preserve local semantic information while incorporating global context, achieving an effective balance between representational capacity and discriminative power. The method integrates seamlessly with existing training and evaluation pipelines, offering a practical and robust alternative to prior approaches, with broad applicability to long-context embedding tasks, multi-vector embeddings, and chunking-based RAG systems.

Limitations

While LMK pooling consistently improves long-context embedding performance, some encoder models (e.g., Ettin encoders) exhibit poor [SEP] token representations, which can cause embedding collapse during LMK training. We find that this can be easily mitigated by using alternative or newly initialized special tokens. LMK is primarily designed for bidirectional encoder models, where it remains robust across diverse architectures. In decoder models, strong causal attention biases often hinder adaptation to bidirectional embedding setups, limiting gains from LMK pooling. Although models such as *Mistral* adapt more effectively (BehnamGhader et al., 2024), improvements are smaller than those observed for encoders, potentially due to extensive long-context pretraining and greater model capacity. Nevertheless, LMK pooling remains broadly applicable due to its consistent effectiveness across architectures and settings.

References

Parul Awasthy, Aashka Trivedi, Yulong Li, Meet Doshi, Riyaz Bhat, Vignesh P, Vishwajeet Kumar, Yushu Yang, Bhavani Iyer, Abraham Daniels, Rudra Murthy, Ken Barker, Martin Franz, Madison Lee, Todd Ward, Salim Roukos, David Cox, Luis Lastras, Jaydeep Sen, and Radu Florian. 2025. Granite embedding r2 models. *Preprint*, arXiv:2508.21085.

Federico Barbero, Alex Vitvitskyi, Christos Perivolaropoulos, Razvan Pascanu, and Petar Velickovic. 2025. Round and round we go! what makes rotary positional encodings useful? In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2Vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*.

Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. 2021. MultiEURLEX - a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6974–6996, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. *CoRR*, abs/2402.03216.

Qian Chen, Wen Wang, Qinglin Zhang, Siqi Zheng, Chong Deng, Hai Yu, Jiaqing Liu, Yukun Ma, and Chong Zhang. 2023. Ditto: A simple and efficient approach to improve sentence embeddings. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5868–5875, Singapore. Association for Computational Linguistics.

Stéphane Clinchant and Florent Perronnin. 2013. Aggregating continuous word embeddings for information retrieval. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 100–109, Sofia, Bulgaria. Association for Computational Linguistics.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *CoRR*, abs/2003.07820.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kenneth C. Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzeminski, Genta Indra Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Diganta Misra, Shreeya Dhakal, Jonathan Rystrøm, Roman Solomatin, Ömer Veysel Çagatan, and 2 others. 2025. MMTEB: massive multilingual text embedding benchmark. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Sofiane Ennadir, Levente Zólyomi, Oleg Smirnov, Tianze Wang, John Pertoft, Filip Cornell, and Lele Cao. 2025. Pool me wisely: On the effect of pooling in transformer-based models. *CoRR*, abs/2510.03339.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Jianfeng Gao, Kristina Toutanova, and Wen-tau Yih. 2011. Clickthrough-based latent semantic models for web search. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, page 675–684, New York, NY, USA. Association for Computing Machinery.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 2333–2338. ACM.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. Nv-embed: Improved techniques for training llms as generalist embedding models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Chaofan Li, Minghao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Defu Lian, Yingxia Shao, and Zheng Liu. 2025a. Making text embedders few-shot learners. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Xiangyang Li, Kuicai Dong, Yi Quan Lee, Wei Xia, Hao Zhang, Xinyi Dai, Yasheng Wang, and Ruiming Tang. 2025b. Coir: A comprehensive benchmark for code information retrieval models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 22074–22091. Association for Computational Linguistics.

Kun Luo, Zheng Liu, Shitao Xiao, Tong Zhou, Yubo Chen, Jun Zhao, and Kang Liu. 2024. Landmark embedding: A chunking-free embedding method for retrieval augmented long-context large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3268–3281, Bangkok, Thailand. Association for Computational Linguistics.

Luke Merrick, Danmei Xu, Gaurav Nuti, and Daniel Campos. 2024. Arctic-embed: Scalable, efficient, and accurate text embedding models. *CoRR*, abs/2405.05374.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

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

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

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. Yarn: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3715–3734. Association for Computational Linguistics.

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

Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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. *CoRR*, abs/2212.03533.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.

Orion Weller, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. 2025. On the theoretical limitations of embedding-based retrieval. *CoRR*, abs/2508.21038.

Shitao Xiao and Zheng Liu. 2022. Retromae v2: Duplex masked auto-encoder for pre-training retrieval-oriented language models. *Preprint*, arXiv:2211.08769.

Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. RetroMAE: Pre-training retrieval-oriented language models via masked auto-encoder. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 538–548, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075, Online. Association for Computational Linguistics.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. *Preprint*, arXiv:2407.19669.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. MIRACL: A multilingual retrieval dataset covering 18 diverse languages. *Trans. Assoc. Comput. Linguistics*, 11:1114–1131.

Dawei Zhu, Liang Wang, Nan Yang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. 2024. LongEmbed: Extending embedding models for long context retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 802–816, Miami, Florida, USA. Association for Computational Linguistics.

Guido Zuccon, Bevan Koopman, Peter Bruza, and Leif Azzopardi. 2015. Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th Australasian Document Computing Symposium*, ADCS '15, New York, NY, USA. Association for Computing Machinery.

A Additional Results

In this section, we present additional results that complement the main findings. In Table 7, we report finetuning results on the MSMarco passage ranking dataset using *gte-en-mlm-base*, analogous to the *ModernBERT-base* results shown in Table 1 in the main text. We observe similar trends, where LMK pooling consistently exhibits strong long context extrapolation, and using variable granularity during finetuning provides added robustness when selecting granularity at inference time. We additionally illustrate the effects of long context extrapolation in Appendix B, showing how LMK performance on LongEmbed datasets improves with increasing context length, while CLS pooling fails to effectively leverage the benefits of longer contexts. Extending to pretraining with RetroMAE, Tables 13 and 14 report the performance of both base models after RetroMAE pretraining with CLS versus LMK pooling, indicating that such pretraining extensions can be readily applied to the proposed pooling mechanism.

We further report results for finetuning on the MSMarco document ranking dataset in Table 8. Here, LMK pooling achieves the best overall in domain performance. For short context benchmarks, CLS pooling attains the highest average score, outperforming LMK by ~ 0.5 points, while mean pooling leads to a notable degradation. In long context evaluations, mean pooling performs best overall, with LMK closely following, whereas CLS pooling lags substantially behind.

We additionally report the full multilingual results on MIRACL, MLDR, and Multi-EURLEX in Tables 10 to 12, corresponding to Table 2 in the main text.

B What does LMK capture?

We examine what LMK pooling learns from token embeddings. LMK pooling partitions an input sequence into fixed size chunks, generating one LMK representation per chunk. Each LMK forms a contextualized representation influenced by the full document. To understand what information LMK embeddings preserve locally, we encode each chunk independently using LMK and compare these non-contextualized embeddings with the contextualized LMK embeddings.

We sample long documents from the MLDR (En) dataset such that each document contains at least 256 chunks, and compute directional Hit@k scores by retrieving the chunk embedding closest to each LMK embedding, as shown in Fig. 3. Since each LMK token is positioned between a left and right chunk, we can measure directional retrieval bias.

We note two key findings. First, each left or right chunk embedding retrieves its corresponding LMK embedding with $\geq 58\%$ Hit@10 accuracy, indicating that these representations preserve substantial local semantic information despite being contextualized by the entire document. Second, LMK embeddings show higher affinity for left neighboring chunks than for right neighbors, indicating a directional bias- they emphasize earlier context more heavily than later context.

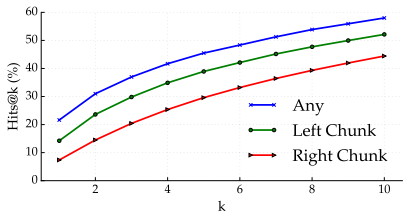


Figure 3: Retrieval Hits@k between LMK token embeddings and left, right, or any (left or right) chunk embeddings for sequences of length 8,192, with LMK granularity 32 ($\geq$248 chunks).

C MTEB-v2 Primarily Evaluates Short-Context Tasks

Although datasets like BEIR and MTEB-v2 spans a diverse set of tasks, the benchmark predominantly consists of short-context datasets and therefore does not meaningfully evaluate long-context modeling, which is the primary setting targeted by LMK pooling. To support this observation, Table 16 reports average document lengths for representative MTEB-v2 datasets. Even under a conservative approximation of $\sim$4 characters per token, the vast majority of datasets remain well below an average sequence length of 512 tokens. Consequently, we do not expect substantial gains over existing pooling strategies on these benchmarks. As discussed throughout the paper, the advantages of LMK pooling are most pronounced on long-context tasks such as MLDR and LongEmbed, while still maintaining competitive performance on shorter-context benchmarks. This also explains the relatively modest differences between pooling methods in the broader MTEB-v2 evaluation reported in Table 15.

Finetuning		Evaluation		In-Domain	Out-of-Domain (Short Context)				Out-of-Domain (Long Context)			
				MSMarco Dev	**BEIR (15)**	**MTEBv2 (10)**	**MIRACL Dev**	**Avg. Short**	**MLDR Test**	**COIR (8)**	**LongEmbed (6)**	**Avg. Long**
Pool	**Gran.**	**Pool**	**Gran.**	NDCG@10	NDCG@10	NDCG@10	NDCG@10		NDCG@10	NDCG@10	P@1 | NDCG@10	
CLS	–	CLS	–	42.3	44.0	44.7	47.9	45.5	27.3	36.6	52.0	38.6
CLS	–	MultiCLS	128	42.8	42.5	42.5	42.6	42.5	31.0	36.9	60.5	42.8
Mean	–	Mean	–	42.1	42.4	43.9	45.6	44.0	31.4	36.6	60.8	42.9
Latent Attn	–	Latent Attn	–	42.0	42.3	43.8	44.9	43.7	30.2	36.8	61.7	42.9
Mean@k	64	Mean@k	64	41.9	43.2	44.2	46.8	44.7	29.3	37.1	56.0	40.8
LMK	32	LMK	32	42.8	44.2	44.9	46.4	45.2	31.0	38.0	63.9	44.3
	64		64	43.0	44.1	44.9	46.3	45.1	24.7	37.6	64.0	42.1
	128		128	43.2	44.2	44.9	47.1	45.4	32.9	37.6	64.8	45.1
	256		256	43.0	44.0	44.6	49.0	45.9	34.6	36.8	59.7	43.7
LMK	Variable	LMK	32	43.0	44.0	44.7	46.7	45.1	27.8	37.7	65.0	43.5
			64	42.9	44.0	44.7	46.4	45.0	29.7	37.5	65.1	44.1
			128	43.0	44.0	44.7	47.0	45.2	30.8	37.5	65.7	44.7
			256	43.0	44.2	44.8	48.1	45.7	31.7	37.6	65.2	44.8
LMK	Sentence	LMK	Sentence	42.6	44.3	45.0	46.9	45.4	33.7	36.6	61.0	43.8

Table 7: Comparison of pooling strategies across in-domain and out-of-domain retrieval benchmarks for *gte-en-mlm-base* when fine-tuned with MSMarco passage data.

Finetuning		Evaluation		In-Domain	Out-of-Domain (Short Context)				Out-of-Domain (Long Context)			
				MSMarco Dev	**BEIR (15)**	**MTEBv2 (10)**	**MIRACL Dev**	**Avg. Short**	**MLDR Test**	**COIR (8)**	**LongEmbed (6)**	**Avg. Long**
Pool	**Gran.**	**Pool**	**Gran.**	NDCG@10	NDCG@10	NDCG@10	NDCG@10		NDCG@10	NDCG@10	P@1 | NDCG@10	
CLS	-	CLS	-	28.7	37.6	39.0	32.3	36.3	29.1	30.1	49.1	36.1
Mean	-	Mean	-	28.3	36.3	37.7	29.0	34.3	30.0	32.5	61.6	41.4
LMK	128	LMK	128	28.6	36.7	37.5	30.1	34.8	30.0	31.6	63.1	41.6
LMK	Variable	LMK	32	28.9	37.1	38.1	32.1	35.8	31.4	30.9	59.8	40.7
			64	28.5	36.9	37.9	31.0	35.3	31.1	30.6	60.4	40.7
			128	28.5	37.1	38.0	31.1	35.4	30.3	30.4	60.0	40.2
			256	28.5	36.9	37.9	30.4	35.1	29.7	30.1	59.9	39.9

Table 8: Comparison of pooling strategies across in-domain and out-of-domain retrieval benchmarks for *gte-en-mlm-base* when fine-tuned with MSMarco document data.

D LMK Pooling on Decoder LLMs

We additionally evaluate LMK pooling on decoder-based embedding models to study whether the benefits observed for bidirectional encoders transfer to large language models (LLMs). Since LMK pooling fundamentally relies on bidirectional interactions between tokens and landmark representations, we restrict our experiments to models that are either explicitly trained or empirically shown to be robust under bidirectional attention. Specifically, we evaluate Mistral-7B-Instruct-v0.3 and LLM2Vec-Meta-Llama-3-8B-Instruct-mntp-unsup-simcse (BehnamGhader et al., 2024), the latter being additionally trained with bidirectional attention and an unsupervised SimCSE objective. In preliminary experiments, we found that strongly causal models such as granite-3.3-8b-instruct and granite-4.1-8b adapt poorly to bidirectional embedding training and therefore cannot effectively leverage LMK pooling. To avoid conflating architectural limitations of causal attention with the pooling mechanism itself, we report results only for models that support stable bidirectional adaptation.

Our training pipeline is built on top of SimLM[2] with modifications enabling bidirectional attention for SDPA-based Transformer implementations and FlashAttention bidirectional kernels where applicable. Training is performed using BF16 precision on 8 NVIDIA H100 80GB GPUs. We optimize models using InfoNCE with temperature $\tau = 0.02$, sharing negatives across devices to increase the effective in-batch negative size. Optimization is performed using DeepSpeed together with HuggingFace Accelerate.

For supervision, we use the BGE-en-icl training mixture with hard negatives and distillation scores provided by Li et al. (2025a), using batch size of 128 with 7 hard negatives for training. Models are trained for 4k steps with a learning rate of

[2] github.com/microsoft/unilm/tree/master/simlm

FINETUNING		EVALUATION		LONG-EMBED RETRIEVAL						
Pool	**Gran.**	**Pool**	**Gran.**	**NarrativeQA** NDCG@10	**QMSum** NDCG@10	**WikiMQA** NDCG@10	**SummScreenFD** NDCG@10	**Needle** P@1	**Passkey** P@1	**Avg.**
CLS	–	CLS	–	30.9	32.3	75.0	82.4	46.5	**82.8**	58.3
Mean	–	Mean	–	**47.1**	**35.7**	**81.1**	86.8	**48.3**	75.8	**62.4**
LMK	Variable	LMK	32	44.4	34.7	79.8	**89.5**	41.8	58.5	58.1

Table 9: Comparison of pooling strategies on LongEmbed retrieval benchmark for *modernbert-base* fine-tuned on MSMarco document data.

FT. Pool	FT. Gran.	Eval. Pool	Eval. Gran.	ara	ben	deu	eng	fas	fin	fra	hin	ind	jpn	kor	rus	spa	swa	tel	tha	yor	zho	avg.
CLS	–	CLS	–	66.1	63.4	47.4	48.1	49.5	68.4	51.6	44.5	48.4	59.2	61.7	58.0	49.2	68.2	75.9	69.1	58.4	50.1	57.6
Mean	–	Mean	–	65.0	61.4	45.7	44.5	50.5	67.3	48.8	43.6	48.2	58.3	61.4	54.7	49.3	68.8	77.3	66.8	50.1	47.2	56.1
Latent Attn	–	Latent Attn	–	65.5	62.9	46.2	45.6	50.6	67.7	49.1	43.2	48.8	58.6	61.0	56.0	50.5	68.9	77.4	67.5	51.2	48.7	56.6
LMK	128	LMK	128	67.2	65.6	45.5	45.2	51.0	68.9	50.8	44.9	50.3	57.5	61.6	56.3	50.2	71.0	78.1	68.9	59.5	52.1	58.0
LMK	Variable	LMK	32	66.5	63.7	45.8	44.7	50.5	69.1	50.0	44.8	50.1	59.9	60.9	57.6	49.8	69.9	77.1	68.9	61.2	49.8	57.8
			64	66.4	63.3	46.2	45.3	50.7	69.0	49.8	44.9	50.1	60.4	61.3	57.7	49.7	69.8	77.1	69.1	61.7	50.2	57.9
			128	66.5	63.8	46.1	44.8	50.9	68.9	50.0	45.2	49.9	59.8	61.1	57.4	50.1	70.1	77.2	68.9	61.5	50.9	57.9
			256	66.5	62.8	46.0	44.8	50.9	68.5	49.4	44.5	49.9	59.8	61.0	57.2	50.0	70.0	77.3	68.8	60.5	51.0	57.7

Table 10: NDCG@10 retrieval performance on MIRACL-HN (18) dataset across languages under different pooling and granularity settings for *mmBERT-base*.

2×10^{-4}, 50 warmup steps, query length capped at 128 tokens, and document length capped at 512 tokens. Following Lee et al. (2025), we prepend task-specific prompts to queries during both training and evaluation.

All evaluations are conducted using MTEB and SentenceTransformers. Short-context evaluation includes MTEBv2 Retrieval, while long-context evaluation includes COIR, LongEmbed, and RULER (Hsieh et al., 2024). We additionally adapt RULER's Needle-in-a-Haystack (NIAH) benchmark into a controlled retrieval setting to better analyze positional bias in embedding models. In the original RULER setup, a model must recover a planted fact (the "needle") from a single long document containing distractor text (the "haystack"). We reformulate this as a retrieval task where the model must identify the correct document from a corpus of C documents, only one of which contains the target needle while the remaining $C - 1$ documents contain distractor content only.

The retrieval query follows the original RULER formulation, asking for the value associated with a specific key, while the inserted needle follows the template: "The special magic {type} for {key} is {value}." We evaluate eight NIAH subtasks spanning varying needle complexity (single-key, multi-key, multi-value, and multi-query variants), six context lengths ranging from 512 to 16K tokens, and ten insertion depths from 10% to 90% of the document. Retrieval performance is measured using Recall@10. This setup enables controlled evaluation of how effectively embedding models localize information distributed across long documents while remaining fully reproducible and independent of held-out test sets.

We first report results on MTEBv2, COIR, and LongEmbed for the two decoder-based embedding models in Appendix D. On Mistral-7B, `LMK` pooling improves MTEBv2 by +1.8 NDCG@10 and LongEmbed by +0.9 points, while remaining competitive with existing pooling strategies on COIR. On LLM2Vec-Llama-3-8B, `LMK` achieves +0.9 improvement on COIR and comparable LongEmbed performance, but shows a -4.1 drop on MTEBv2. In contrast, EOS pooling exhibits a substantial degradation, likely because the model was adapted for bidirectional training using mean-pooled representations. Overall, these results suggest that `LMK` pooling remains competitive for decoder-based embedding models, although the gains diminish with decoder LLMs than those observed for encoder models. We hypothesize this is due to two factors: (1) decoder LLMs are primarily pretrained with causal attention and are therefore harder to adapt to bidirectional embedding settings, and (2) large-scale long-context pretraining already provides stronger extrapolation capabilities compared to encoder models.

To further analyze pooling behavior under controlled long-context settings, we evaluate different pooling strategies on RULER NIAH retrieval tasks, with results shown in Figs. 5 to 12. In these experiments, we vary the maximum sequence length

FT. Pool	FT. Gran.	Eval. Pool	Eval. Gran.	ara	cmn	deu	eng	fra	hin	ita	jpn	kor	por	rus	spa	tha	avg.
CLS	–	CLS	–	21.9	9.0	13.6	8.5	42.1	13.1	35.7	28.5	13.3	50.9	35.3	45.0	7.2	24.9
Mean	–	Mean	–	11.9	7.4	14.3	6.0	7.9	11.2	15.3	30.2	5.4	36.3	26.5	39.8	9.4	17.0
Latent Attn	–	Latent Attn	–	15.5	6.0	18.0	4.3	11.8	11.3	20.9	30.8	6.3	41.5	30.9	46.2	11.1	19.6
LMK	128	LMK	128	31.0	15.5	31.5	24.7	60.9	19.3	48.3	36.8	21.2	63.4	46.4	60.6	23.5	37.2
LMK	Variable	LMK	32	30.8	16.9	33.2	27.1	60.3	20.2	51.8	39.3	25.2	63.7	48.2	63.1	23.8	38.7
			64	31.0	16.3	32.8	27.0	60.4	19.5	51.8	38.6	25.4	63.9	47.9	63.6	23.6	38.6
			128	30.4	15.6	32.3	27.2	60.9	18.5	52.1	39.5	24.0	63.1	47.7	63.4	23.7	38.3
			256	30.2	14.3	32.4	27.4	60.7	18.2	51.8	39.8	23.1	64.0	46.2	63.6	21.8	38.0

Table 11: NDCG@10 retrieval performance on MLDR (13) dataset across languages under different pooling and granularity settings for *mmBERT-base*.

FT. Pool	FT. Gran.	Eval. Pool	Eval. Gran.	deu	eng	bul	por	lit	swe	est	spa	nld	pol	fra	slk	fin	mlt	ita	hun	lav	ces	ell	slv	dan	ron	hrv	avg.
CLS	–	CLS	–	27.2	26.8	28.7	27.3	29.6	26.8	27.6	26.0	26.4	28.5	27.7	27.3	27.1	24.6	27.3	29.1	30.5	30.4	25.1	26.9	27.6	29.3	25.2	27.5
Mean	–	Mean	–	27.3	27.9	30.3	27.0	29.0	26.9	28.7	26.9	25.8	28.6	26.5	27.7	26.5	24.9	26.7	28.8	30.6	29.9	25.9	27.4	26.3	29.2	23.5	27.5
Latent Attn	–	Latent Attn	–	28.1	27.3	30.8	27.4	29.3	27.2	29.2	27.6	27.4	28.5	26.9	28.2	27.3	25.1	27.2	30.0	31.8	30.3	26.4	28.3	26.7	29.2	24.0	28.0
LMK	128	LMK	128	31.5	30.0	33.5	29.4	33.1	31.3	30.3	30.5	30.3	33.1	31.9	31.8	31.3	29.5	30.3	33.4	33.9	34.2	28.5	31.6	31.2	33.6	29.8	31.5
LMK	Variable	LMK	32	34.1	32.1	34.9	31.6	34.8	32.9	32.8	32.3	33.2	35.5	34.0	33.8	33.7	32.5	32.5	35.4	35.4	36.7	30.3	34.2	33.4	35.2	32.6	33.7
			64	34.0	31.8	34.9	31.3	34.7	32.6	32.6	32.1	32.8	35.3	33.8	33.8	33.1	32.2	32.3	35.1	35.4	36.4	30.3	33.9	33.0	35.0	32.1	33.4
			128	34.1	31.6	34.8	31.2	34.7	32.5	32.4	31.8	32.6	35.2	33.8	33.7	32.8	31.9	32.4	34.8	35.6	36.0	30.2	33.7	32.7	34.8	31.3	33.2
			256	34.0	31.5	34.5	30.9	34.4	32.3	32.4	31.5	32.4	35.1	33.4	33.4	32.5	31.4	32.3	34.5	35.3	35.7	29.9	33.2	32.3	34.6	30.9	33.0

Table 12: Zero-shot long-document classification F1 scores on MultiEURLEX across 23 languages using fine-tuned *mmBERT-base*, evaluated with a maximum sequence length of 8192.

from 512 to 16K tokens and move the needle position from 0–90% depth to study positional bias during encoding. EOS/last-token pooling exhibits a strong bias toward later positions, achieving its best performance when the needle appears near the end of the sequence and degrading substantially as the needle shifts toward earlier positions. Mean pooling shows lower positional bias but consistently underperforms LMK pooling. In contrast, LMK pooling produces more stable retrieval performance across varying sequence lengths and needle depths, indicating improved robustness to positional variation in long documents. While these results demonstrate the potential of LMK pooling for decoder-based embedding models, we believe further investigation is needed to better understand its interaction with causal pretraining and bidirectional adaptation, which we leave to future work.

E Training and Evaluation Hyperparameters

Our experiments are divided into English and Multilingual settings, with both short- and long-context training and evaluation.

English: We use Transformer-based text encoders supporting long-context training, namely *gte-en-mlm-base* and *ModernBERT-base*. Models are trained on MS MARCO passage and document ranking datasets using hard negatives (7 for passage, 1 for document) and distillation scores provided by Li et al. (2025a). Training is performed for 5k steps with a learning rate of 2×10^{-5}, warmup steps of 250, and query maximum length capped at 128 tokens. Passage ranking uses an effective query batch size of 2,048 and maximum sequence length of 512 tokens; document ranking uses batch size 256 and sequences up to 8,192 tokens.

Multilingual: We use *mmBERT-base* and *gte-multilingual-base*. Models are fine-tuned on multilingual datasets from Chen et al. (2024) with hard negatives (7) and distillation scores. Training is performed for 10k steps with learning rate 2×10^{-5}, batch size 1,024, and maximum sequence length 512 tokens. For ablations with sequence lengths up to 8,192 tokens, batch size is reduced to 64 due to memory constraints.

Training setup: We build our training pipeline on top of FlagEmbedding. We use BF16 precision on 8 NVIDIA H100 80GB GPUs. InfoNCE loss temperature is set to $\tau = 0.02$, and negatives are shared across devices to increase effective in-batch negative size. Training is optimized using DeepSpeed and HuggingFace Accelerate.

Evaluation: All embedding evaluations are conducted using MTEB and SentenceTransformers. For English, short-context benchmarks include BEIR-15, MTEBv2, and MIRACL Retrieval; long-context benchmarks include MLDR, COIR, and LongEmbed. For multilingual experiments, short-

Pretraining		Finetuning		Evaluation		In-Domain		Out-of-Domain (Short Context)				Out-of-Domain (Long Context)			
						MSMarco Dev		BEIR (15)	MTEBv2 (10)	MIRACL-HN EN	Avg. Short	MLDR EN	COIR (8)	LongEmbed (6)	Avg. Long
Pool	Gran.	Pool	Gran.	Pool	Gran.	NDCG@10	MRR@10	NDCG@10	NDCG@10	NDCG@10		NDCG@10	NDCG@10	P@1 | NDCG@10	
CLS	–	CLS	–	CLS	–	42.5	36.1	44.7	45.5	47.4	45.9	31.2	42.1	60.8	44.7
LMK	64	LMK	64	LMK	64	42.6	36.2	44.5	45.5	46.8	45.6	36.7	43.0	66.1	48.6
	128		128		128	42.8	36.5	44.2	44.8	47.3	45.4	34.6	43.2	65.7	47.8
	256		256		256	42.8	36.3	44.5	45.1	48.7	46.1	36.0	42.8	67.6	48.8
LMK	Variable	LMK	Variable	LMK	32	42.6	36.1	44.3	45.4	46.7	45.5	37.8	43.5	68.4	49.9
					64	42.5	36.2	44.3	45.3	46.5	45.4	38.1	43.3	68.4	49.9
					128	42.7	36.3	44.4	45.4	47.1	45.6	37.5	43.2	68.1	49.6
					256	42.6	36.2	44.5	45.4	48.4	46.1	36.6	43.3	67.7	49.2
LMK	Sentence	LMK	Sentence	LMK	Sentence	42.5	35.9	44.5	45.3	47.4	45.7	36.2	42.2	63.6	47.3

Table 13: Comparison of pooling strategies on in-domain and out-of-domain retrieval benchmarks for *gte-en-mlm-base*, pretrained with CLS or LMK pooling and fine-tuned on MS MARCO passage data.

Pretraining		Finetuning		Evaluation		In-Domain	Out-of-Domain (Short Context)				Out-of-Domain (Long Context)			
						MSMarco Dev	BEIR (15)	MTEBv2 (10)	MIRACL-HN EN	Avg. Short	MLDR EN	COIR (8)	LongEmbed (6)	Avg. Long
Pool	Gran.	Pool	Gran.	Pool	Gran.	NDCG@10	NDCG@10	NDCG@10	NDCG@10		NDCG@10	NDCG@10	P@1 | NDCG@10	
CLS	–	CLS	–	CLS	–	39.8	42.8	44.4	47.9	45.0	24.9	44.3	42.1	37.1
LMK	64	LMK	32	LMK	32	40.3	42.5	44.1	47.7	44.8	31.9	46.8	59.8	46.2
LMK	Variable	LMK	Variable	LMK	32	40.2	41.3	42.5	47.8	43.9	31.8	46.2	66.4	48.1
LMK	Sentence	LMK	Sentence	LMK	Sentence	40.0	42.0	43.2	46.9	44.0	32.3	43.4	64.9	46.9

Table 14: Comparison of pooling strategies on in-domain and out-of-domain retrieval benchmarks for *ModernBERT-base*, pretrained with CLS or LMK pooling and fine-tuned on MS MARCO passage data.

context evaluation uses MIRACL Hard Negatives (18 languages) and long-context evaluation uses MLDR (13 languages) and LongEmbed. Zero-shot long-document classification is evaluated on Multi-EURLEX (23 languages) using Macro-F1.

F Dataset Information

We begin with standard English retrieval training datasets commonly used in dense retrieval, such as MSMarco Passage Ranking, TREC-DL, and Natural Questions (NQ) (Nguyen et al., 2016; Craswell et al., 2020; Kwiatkowski et al., 2019). These datasets have been instrumental in training effective short-context retrievers; however, they primarily consist of relatively short passages and therefore do not expose models to long document structures during training.

To partially address this limitation, the MSMarco Document Ranking dataset is often used, as it contains longer documents compared to passage-level data. However, we find that this dataset remains limited in terms of true long-context coverage, with document lengths falling well below those encountered in many real-world retrieval scenarios (Fig. 13).

More recently, embedding training datasets have diversified substantially. Modern training mixtures, such as those used in the *bge-en-icl* framework (Li et al., 2025a), combine heterogeneous data sources spanning multiple tasks, including semantic textual similarity, clustering, retrieval, classification, question answering, and natural language inference. Statistics for these datasets are reported in Table 18, where we show the number of samples and character length distributions, including the median, 25th, and 75th percentiles. Even under a conservative estimate of four to five characters per token, the majority of these datasets have median sequence lengths of at most 250 tokens. The MSMarco Document dataset remains an exception, yet even at the 75th percentile its length is typically below 2000 tokens. This highlights that, despite the increased diversity of modern embedding training corpora, long-context supervision remains scarce in English retrieval training data.

Training directly on long-context inputs is further constrained by practical considerations, including increased computational cost, reduced batch sizes, and longer training times. **LMK pooling mitigates these limitations by enabling strong long-context extrapolation even when trained predominantly on short-context data**, as demonstrated consistently across our experimental results.

For evaluation, we first report in-domain performance on the MSMarco Dev set and out-of-domain performance on the BEIR-15 benchmark, which is

Model	FINETUNE		EVAL		Class.	Clust.	PairCls.	Rerank.	Retr.	STS	Summ.	Avg.
	Pool	Gran.	Pool	Gran.	(8)	(8)	(3)	(2)	(10)	(9)	(1)	(41)
gte-en-mlm-base	CLS	–	CLS	–	69.5	41.6	83.3	44.8	44.7	77.1	27.6	58.5
	Mean	–	Mean	–	70.1	40.9	83.3	44.5	44.9	75.9	30.4	58.3
	LMK	256	LMK	256	70.0	41.5	83.3	44.9	44.6	76.5	27.2	58.4
ModernBERT-base	CLS	–	CLS	–	69.5	41.6	80.4	44.1	44.2	76.0	24.2	57.8
	Mean	–	Mean	–	69.6	41.2	79.4	43.6	43.3	75.3	24.8	57.3
	LMK	32	LMK	32	69.9	42.2	80.2	44.3	45.9	74.9	25.6	58.1
	LMK	Variable	LMK	128	69.9	42.5	80.5	44.1	45.1	75.2	25.7	58.1

Table 15: MTEB-v2 embedding benchmark results. We report task-wise averages across pooling strategies for *gte-en-mlm-base* and *ModernBERT-base*, trained on MSMarco Passage data.

Dataset	Avg. Char. Length
STS22.v2	3372
SummEvalSummarization.v2	2100
Touche2020Retrieval.v3	2096
MedrxivClusteringP2P.v2	1981
BiorxivClusteringP2P.v2	1664
ImdbClassification	1294
StackExchangeClusteringP2P.v2	1091
ArXivHierarchicalClusteringP2P	1008
ArguAna	830
FiQA2018	760
ClimateFEVERHardNegatives	643
TRECCOVID	637
SCIDOCS	596
CQADupstackUnixRetrieval	497
FEVERHardNegatives	352
ToxicConversationsClassification	289
CQADupstackGamingRetrieval	241
HotpotQAHardNegatives	187
BIOSSES	155
MedrxivClusteringS2S.v2	115
AmazonCounterfactualClassification	107
TwitterURLCorpus	80
ArXivHierarchicalClusteringS2S	73
SprintDuplicateQuestions	68
TweetSentimentExtractionClassifica	68
SprintDuplicateQuestions	65
STSBenchmark	64
STS12	64
STS13	62
STS15	58

Table 16: Average document character lengths for representative MTEB-v2 datasets. Even under a conservative approximation of $\sim$4 characters per token, most datasets correspond to relatively short contexts.

MODEL		MTEBv2	COIR	LONGEMBED	
Base	Pool	NDCG@10	NDCG@10	P@1	NDCG@10
Mistral-7B-Inst	Mean	49.4	57.8	68.8	
	EOS	49.7	**58.8**	68.3	
	LMK	**51.2**	58.5	**69.7**	
LLM2Vec-Llama3-8b	Mean	**46.9**	56.5	34.8	
	EOS	8.5	9.3	19.7	
	LMK	42.8	**57.4**	**35.0**	

Table 17: Comparison of pooling strategies for decoder-based embedding models across short- and long-context retrieval benchmarks.

widely used for assessing retrieval generalization. We additionally evaluate on the MTEB-v2 retrieval subset, consisting of ten retrieval datasets designed to enable faster yet representative benchmarking (Enevoldsen et al., 2025), as well as the MIRACL English Dev set, which is constructed from human annotated relevance judgments over Wikipedia articles. Since these benchmarks primarily consist of short documents (most of which have of at most 512 tokens), they are insufficient for evaluating long-context retrieval behavior. To address this limitation, we further include the MLDR English Test set, the COIR code retrieval benchmark spanning diverse programming tasks and languages, and the LongEmbed benchmark, which focuses on extremely long documents with an average length exceeding 5.5k words. Together, these datasets enable a comprehensive evaluation of retrieval performance under long-context settings, including sequence lengths beyond 32k tokens.

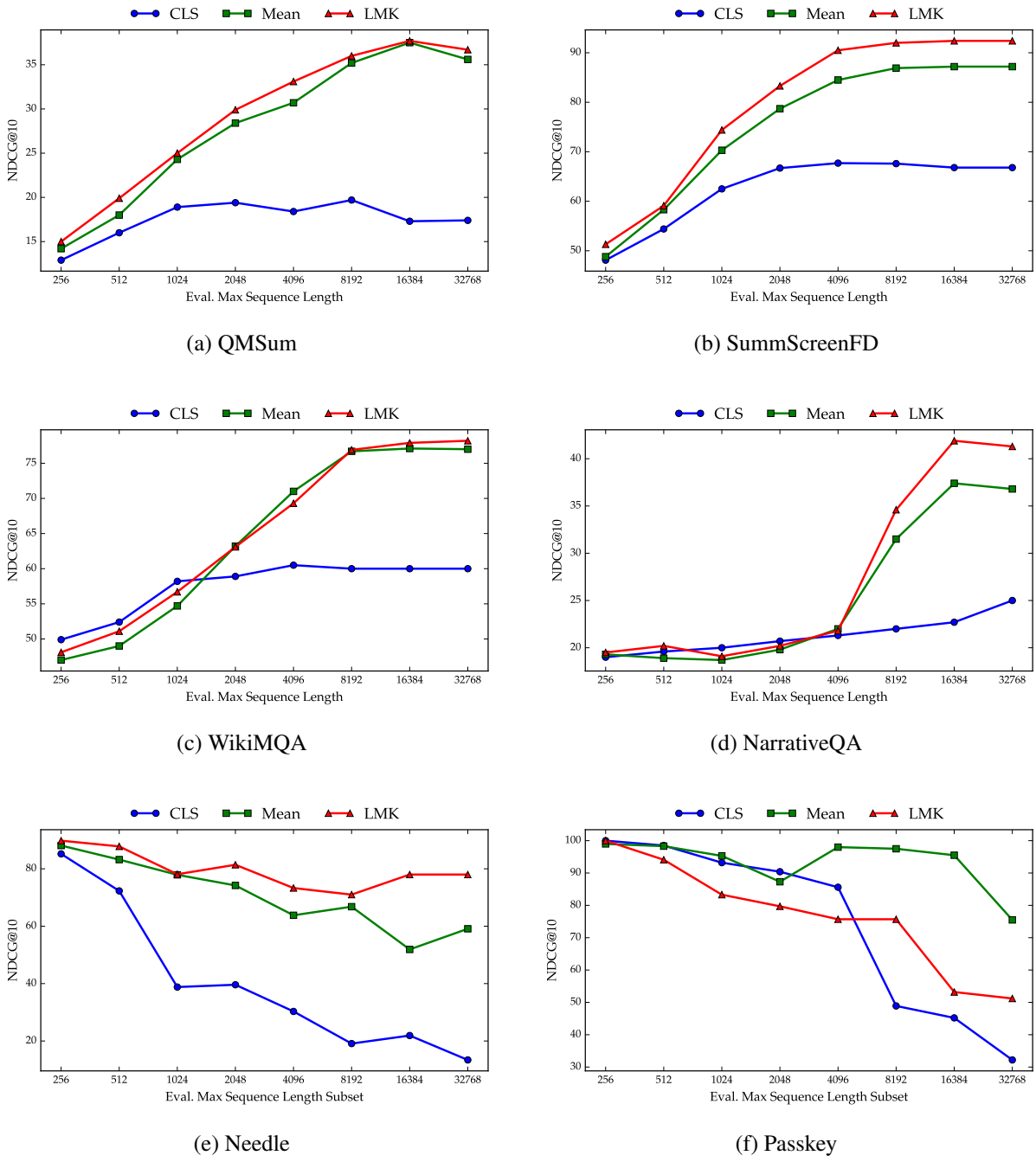
For multilingual training, we use the *BGE-m3* training data introduced by Chen et al. (2024), which comprises a diverse collection of datasets spanning a wide range of languages, as summarized in Table 19. This mixture includes the MLDR training set, which contains substantially longer documents than MSMarco Document Ranking and is therefore used for our long-context training experiments. However, MLDR constitutes only about 42k query document pairs out of the roughly three million total training pairs, limiting its overall influence during training and resulting in relatively weak long-context extrapolation when using conventional pooling methods.

For multilingual evaluation, we use all 18 languages available in the MIRACL Dev set for short-context tasks and the MLDR 13-language test set for long-context tasks. Since the corresponding training sets were already seen during model training, we additionally evaluate out-of-domain long-

context performance. To this end, we include the Multi-EURLEX dataset, which contains 65k EU legal documents in 23 languages labeled with EUROVOC concepts by the EU Publication Office. For this out-of-domain evaluation, we report the F1 score. Finally, we include LongEmbed as a long-context benchmark to assess the models' ability to handle long-context retrieval tasks.

Dataset	Count	Median	P25	P75
amazon counterfactual classification	3,907	80	80	80
amazon reviews classification	20,000	78	67	78
arXiv abstract	20,000	825	581	1,135
arguana	4,065	813	595	1,098
arxiv title	20,000	66	50	86
banking classification	10,003	35	27	41
biorxiv abstract	4,070	1,614.5	1,325	1,975.75
biorxiv title	4,070	102	83	125
eli5	325,475	393	232	621
emotion classification	14,341	46	45	49
fever	29,096	1,633	859	2,496
fiqa	5,500	870.5	503	1,453.25
hotpotqa	84,516	396	270	559
imdb classification	25,000	57	57	57
medrxiv abstract	1,157	1,836	1,440	2,271
medrxiv title	1,157	113	90	141
msmarco document	367,013	3,897	2,160	7,758
msmarco passage	485,823	336	281	454
mtop intent classification	15,667	18	16	26
nli	275,601	37	27	55
nq	58,568	624	599	652
quora	60,202	47	37	61
reddit clustering	20,000	49	32	70
reddit clusteringP2P	20,000	493	306	850.25
scidocsrr	12,655	1,006	753	1,299
squad	87,599	693	559	895
stack exchange clustering	20,000	46	31	64
stack exchange clusteringP2P	20,000	649	407	1,049
stack overflow dup questions	19,847	47	36	60
sts	2,006	56	39	122
toxic conversations classification	25,000	26	26	26
trivial	60,315	625	601	652
tweet sentiment extraction classification	27,481	114	110	114
twenty news groups	11,854	28	20	39

Table 18: Character length statistics for positive documents in the *bge-en-icl* dataset, used to train English dense retrievers by Li et al. (2025a). We report the number of samples and the median, 25th (P25), and 75th (P75) percentiles. Assuming $\sim$4–5 characters per token, most datasets correspond to sequence lengths well below those required for learning robust long-context representations.



(a) QMSum

(b) SummScreenFD

(c) WikiMQA

(d) NarrativeQA

(e) Needle

(f) Passkey

Figure 4: Long-context retrieval performance across datasets for different pooling strategies using *ModernBERT-base* fine-tuned on MSMarco Passages.

Dataset	Count	Median	P25	P75
DuReader	80,416	296	258	411
HotpotQA	84,516	396	270	559
Law	2,054	607	356.5	1455.75
MIRACL (ar)	3,495	472	302	713.5
MIRACL (bn)	1,631	581	405.5	831
MIRACL (en)	2,863	661	446	901
MIRACL (es)	2,162	544.5	376	758
MIRACL (fa)	2,107	408	256	624
MIRACL (fi)	2,897	466	332	651
MIRACL (fr)	1,143	509	320	769.5
MIRACL (hi)	1,169	500	326	754
MIRACL (id)	4,071	607	413	836
MIRACL (ja)	3,477	227	153	324
MIRACL (ko)	868	228	156	332
MIRACL (ru)	4,683	550	360	821
MIRACL (sw)	1,901	278	180	428
MIRACL (te)	3,452	566	395	946
MIRACL (th)	2,972	485	339	691
MIRACL (zh)	1,312	168	114	246
MLDR (ar)	1,817	13,535	5,174	21,921
MLDR (de)	1,847	24,850	15,070	29,238.5
MLDR (en)	10,000	18,229	10,553.75	25,636.25
MLDR (es)	2,254	22,453.5	6,579.25	28,463
MLDR (fr)	1,608	23,685	13,130.75	27,501.25
MLDR (hi)	1,618	12,323.5	5,759.75	20,120
MLDR (it)	2,151	25,551	12,454.5	29,046.5
MLDR (ja)	2,262	9,311.5	4,068.25	11,003.75
MLDR (ko)	2,198	7,529.5	2,563.25	11,906.5
MLDR (pt)	1,845	25,574	12,549	29,334
MLDR (ru)	1,864	18,851	7,516.75	23,289.5
MLDR (th)	1,970	10,283.5	6,651.25	22,316.5
MLDR (zh)	10,000	6,079.5	3,610	8,624.75
MSMARCO	485,905	336	281	454
Mr. TyDi	48,729	434	277	647
NQ	58,568	624	599	652
SQuAD	87,599	693	559	895
T2Ranking	90,467	316	180	604
Trivia	60,315	625	601	652
cMedQAv2	50,000	90	62	125
EN-NLI	274,951	37	27	55
mMARCO	100,000	110	84	152
ZH-NLI (ATEC)	11,325	12	10	15
ZH-NLI (BQ)	12,599	10	7	14
ZH-NLI (LCQMC)	10,000	10	8	12
ZH-NLI (PAWSX)	10,000	42	32	55
ZH-NLI (QBQTC v2)	10,000	22	14	31
ZH-NLI (STS)	249	15	11	19
ZH-NLI (AFQMC)	10,534	12	10	16

Table 19: Character length statistics for positive documents across *bge-m3* dataset, used by Chen et al. (2024). We report the number of samples along with the median, 25th (P25), and 75th (P75) percentiles. MLDR datasets feature substantially longer sequences than standard retrieval benchmarks, and without them, models using CLS or mean pooling struggle to generalize to long-context settings.

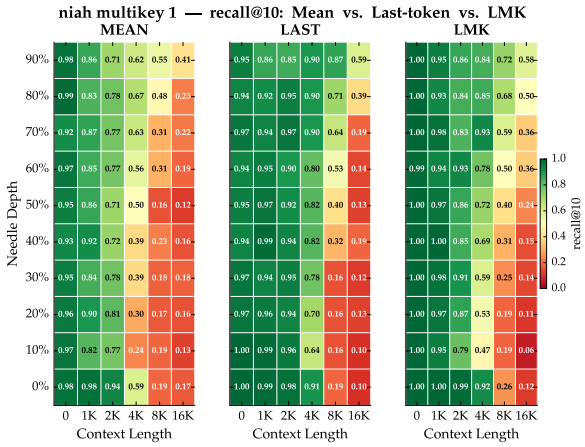


Figure 5: Recall@10 comparison of pooling strategies on Mistral-7B across varying context lengths (512–16K) and needle depths (10%–90%) on the RULER NIAH-Multikey-1 retrieval.

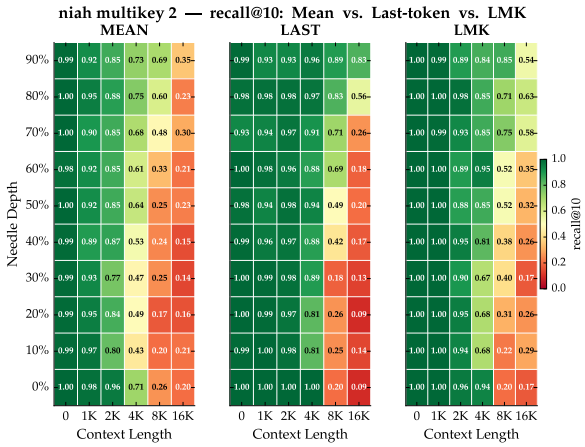


Figure 6: Recall@10 comparison of pooling strategies on Mistral-7B across varying context lengths (512–16K) and needle depths (10%–90%) on the RULER NIAH-Multikey-2 retrieval.

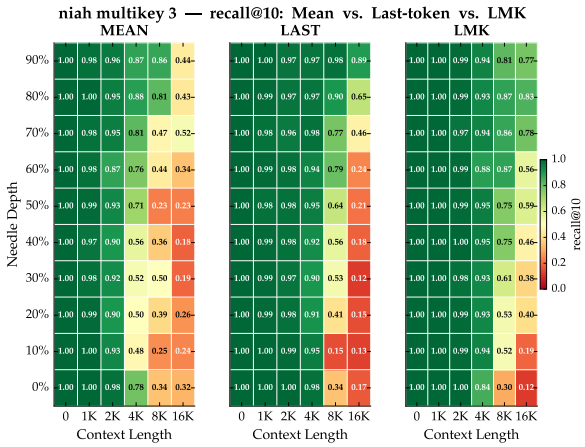


Figure 7: Recall@10 comparison of pooling strategies on Mistral-7B across varying context lengths (512–16K) and needle depths (10%–90%) on the RULER NIAH-Multikey-3 retrieval.

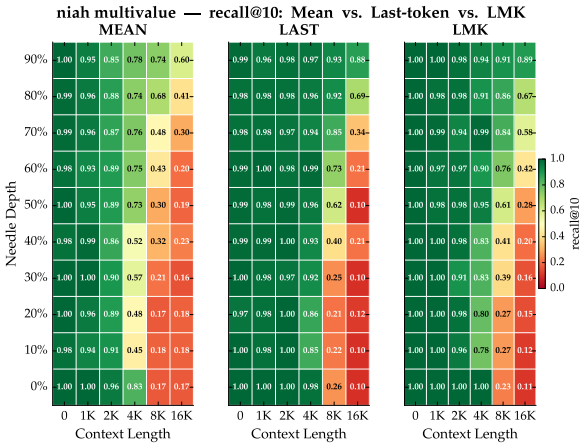


Figure 8: Recall@10 comparison of pooling strategies on Mistral-7B across varying context lengths (512–16K) and needle depths (10%–90%) on the RULER NIAH-Multivalue retrieval.

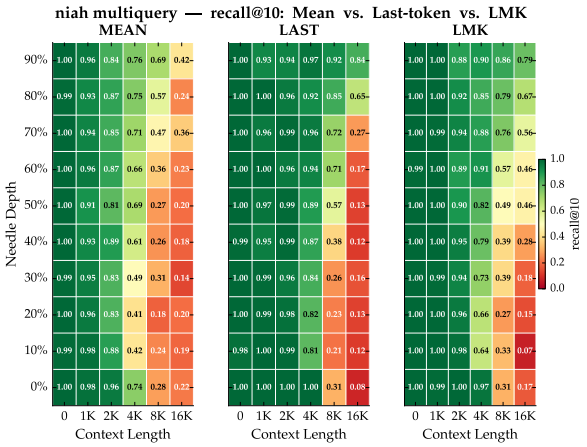


Figure 9: Recall@10 comparison of pooling strategies on Mistral-7B across varying context lengths (512–16K) and needle depths (10%–90%) on the RULER NIAH-Multiquery retrieval.

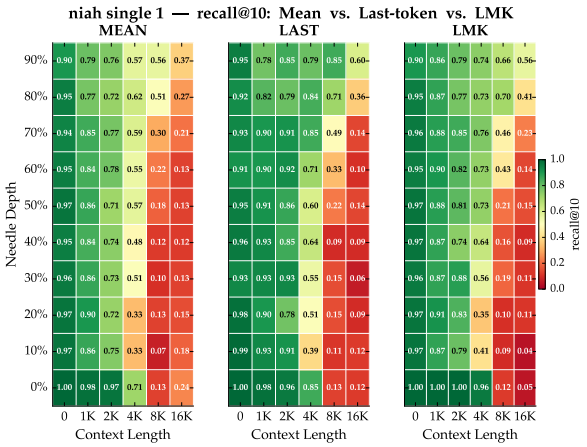


Figure 10: Recall@10 comparison of pooling strategies on Mistral-7B across varying context lengths (512–16K) and needle depths (10%–90%) on the RULER NIAH-Single-1 retrieval.

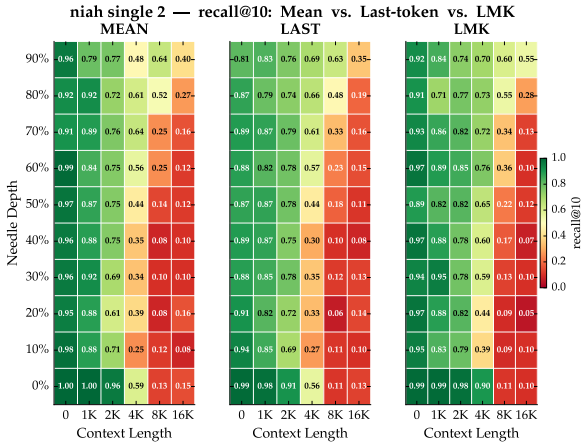


Figure 11: Recall@10 comparison of pooling strategies on Mistral-7B across varying context lengths (512–16K) and needle depths (10%–90%) on the RULER NIAH-Single-2 retrieval.

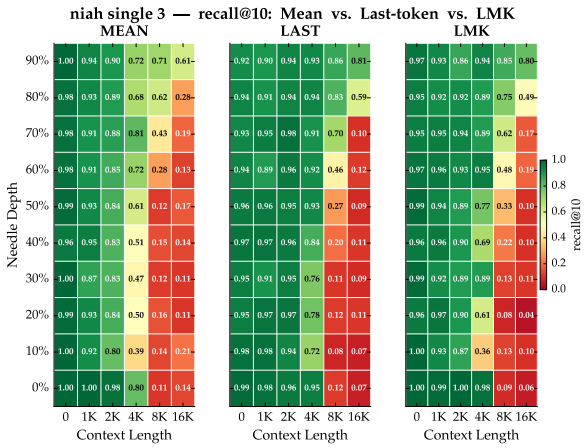


Figure 12: Recall@10 comparison of pooling strategies on Mistral-7B across varying context lengths (512–16K) and needle depths (10%–90%) on the RULER NIAH-Single-3 retrieval.

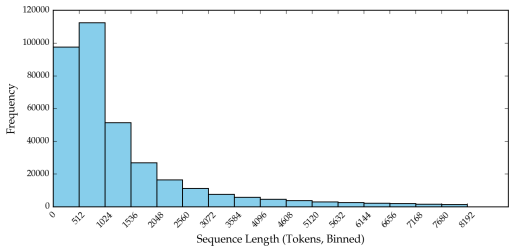


Figure 13: Binned distribution of passage token lengths in the MS MARCO Document Ranking training set, computed using the *ModernBERT-base* tokenizer.