Learning to Compress: Unlocking the Potential of Large Language Models for Text Representation

Yeqin Zhang[1,2], Yizheng Zhao[1,2], Chen Hu[3], Binxing Jiao[3], Daxin Jiang[3], Ruihang Miao[3]*, Cam-Tu Nguyen[1,2]*

[1] State Key Laboratory for Novel Software Technology, Nanjing University, China
[2] School of Artificial Intelligence, Nanjing University, China
[3] Stepfun, China
zhangyeqin@smail.nju.edu.cn, zhaoyz@nju.edu.cn
{hatcher, benjiao, djiang, miaoruihang}@stepfun.com, ncamtu@nju.edu.cn

Abstract

Text representation plays a critical role in tasks like clustering, retrieval, and other downstream applications. With the emergence of large language models (LLMs), there is increasing interest in harnessing their capabilities for this purpose. However, most of the LLMs are inherently causal and optimized for next-token prediction, making them suboptimal for producing holistic representations. To address this, recent studies introduced pretext tasks to adapt LLMs for text representation. Most of these tasks, however, rely on token-level prediction objectives, such as the masked next-token prediction (MNTP) used in LLM2Vec. In this work, we explore the untapped potential of context compression as a pretext task for unsupervised adaptation of LLMs. During compression pre-training, the model learns to generate compact memory tokens, which substitute the whole context for downstream sequence prediction. Experiments demonstrate that a well-designed compression objective can significantly enhance LLM-based text representations, outperforming models trained with token-level pretext tasks. Further improvements through contrastive learning produce a strong representation model (LLM2Comp) that outperforms contemporary LLM-based text encoders on a wide range of tasks while being more sample-efficient, requiring significantly less training data.

Code — https://github.com/longtaizi13579/LLM2Comp
Extended version — https://arxiv.org/abs/2511.17129

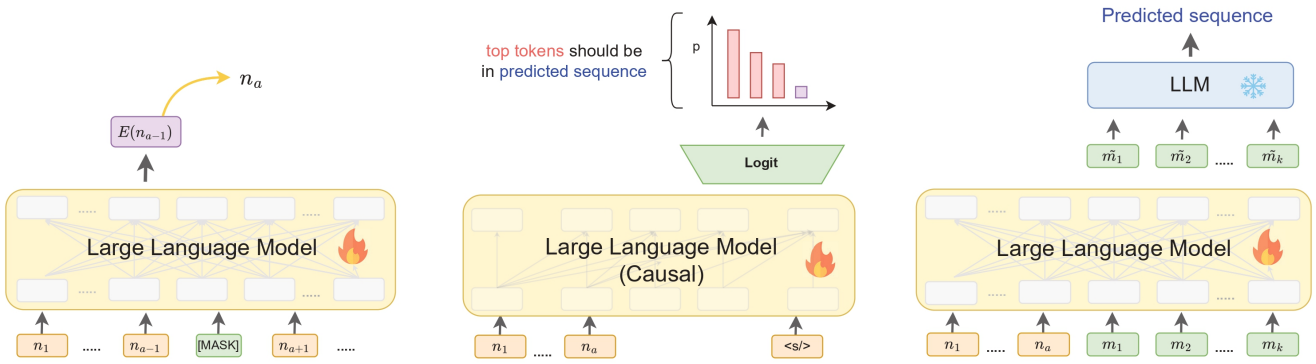
1 Introduction

Text embedding models, which transform semantic content into vector representations, are fundamental to numerous tasks such as retrieval and recommendation. Early methods like TF-IDF and BM25 relied on simple statistics, thus falling short in capturing sequence semantics. The advent of deep neural networks led to the development of techniques such as Word2Vec (Mikolov et al. 2013). This paradigm shift paved the way for the foundational models like BERT (Devlin et al. 2019) and T5 (Raffel et al. 2020).

Recently, there has been growing interest in leveraging the powerful capabilities of LLMs for text representation. However, most LLMs are inherently causal and optimized for next-token prediction, which makes them inherently suboptimal for generating holistic, coherent representations of entire sequences. To address this limitation, recent studies have proposed various pretext tasks for unsupervised adaptation of LLMs, focusing primarily on token-level prediction objectives. For example, as shown in Figure 1a, LLM2Vec (BehnamGhader et al. 2024) first transforms the causal attention mechanism of LLMs into a bidirectional form, and then adopts masked next token prediction (MNTP) as a pretext task to align the training objectives of causal LLMs with those of bidirectional models such as BERT. In this setup, MNTP randomly masks tokens within a sentence and leverages contextual representations of preceding tokens to predict masked ones. In contrast, as shown in Figure 1b, Llama2Vec (Li et al. 2024) employs two pretext tasks, Embedding-Based Auto-Encoding (EBAE) and Embedding-Based Auto-Regression (EBAR), which predict tokens within the original sequence or the continued sequence. Although such objectives can capture the "bag-of-tokens" information, they cannot fully preserve the coherent semantic integrity of the entire sequence. These tasks, therefore, remain fundamentally token-level rather than sequence-level prediction.

In this work, we explore context compression (Figure 1) as a pretext task for the unsupervised adaptation of LLMs to text encoders. Specifically, during compression pre-training, the model learns to produce compact "memory tokens" that replace the original context for downstream sequence prediction tasks. The conceptual difference between our objectives and previous pretext tasks is illustrated in Figure 1. Here, we explore several compression objectives: *reconstruction task* (Ge et al. 2024; Xu et al. 2024; Cheng et al. 2024) involves regenerating original sentences; *continuation task* (Ge et al. 2024; Chevalier et al. 2023; Qin et al. 2024; Xu et al. 2024; Shao et al. 2024) focuses on generating correct subsequent tokens. Our preliminary experiment shows that the reconstruction task does not provide satisfactory results, whereas the continuation task trained with NLL loss (CT-NLL) suffers from unstable training. Inspired by (Mu, Li, and Goodman 2023; Wingate, Shoeybi, and Sorensen 2022), we then propose *Continuation Task with Knowledge Distillation* (CTKD) as a pretext task, which aims to predict the probability of subsequent tokens in alignment with the original extended

*Corresponding authors.



(a) LLM2Vec (b) Llama2Vec (c) LLM2Comp

Figure 1: The comparison of different pretext tasks: (1) MNTP (Mask Next Token Prediction) in LLM2Vec; (2) EBAE or EBAR in Llama2Vec; (3) Context Compression task in LLM2Comp (ours)

sequence from the soft prompt. Our experiments demonstrate that this well-designed objective (CTKD) can significantly enhance LLM-based text representations.

We then performed a comprehensive analysis and found that LLMs' embeddings trained with the compression pretext task still suffer from dimensional collapse (Jing et al. 2022), where the vectors lie in a low-dimensional subspace instead of using the full embedding space. However, compression pretraining with the CTKD objective is less affected by this issue than CT-NLL. This likely explains the advantage of CTKD over the CT-NLL objective. To further address dimensional collapse, we apply contrastive post-training on the model pretrained with CTKD. This post-training consists of an unsupervised contrastive phase followed by supervised contrastive learning (SCL). In particular, SCL encourages the embedding vectors of the negative samples to be pulled away, reducing the dimensional collapse and leading to significant performance gains. Our experiments show that CTKD pretraining synergizes with contrastive learning. As a result, our model, LLM2Comp, outperforms other LLM-based models on many MTEB benchmark tasks. LLM2Comp is also more sample-efficient, requiring much less (supervised) data during the contrastive learning phases than competing methods.

Our key contributions are summarized as follows:

- We thoroughly investigate the untapped potential of compression pretraining for adapting LLMs to text representation tasks. Through empirical analysis, we provide insights into the crucial factors for the success of such a pretraining task, including the optimal training objective (CTKD) and the appropriate number of memory tokens.
- We delve into the reasons behind the advantage of CTKD over other compression objectives, demonstrating that it is less prone to the dimensional collapse issue and thus more suited for downstream text representation.
- We show that further enhancements through contrastive learning help alleviate the dimensional collapse issue. Note that the CTKD task provides a robust foundation for text representation, enabling our model, LLM2Comp, to significantly outperform contemporary models such as LLM2Vec and Llama2Vec with less training data.

2 Related Works

Recent methods (Nie et al. 2024) for adapting LLMs to text embedders can be broadly classified into training-free and training-based approaches.

Training-free Methods The simple way to use LLMs as text encoders is to take the last token's hidden state of the final transformer layer as the representation. This is known as the last token pooling mechanism. However, the last token's representation is optimized for next-token prediction rather than for aggregating global embedding. Recently, several prompting strategies such as PromptEOL (Jiang et al. 2024) and MetaEOL (Lei et al. 2024) have been proposed to enhance the representational ability of the last token. An alternative approach is weighted mean pooling (Muennighoff 2022), which aggregates information from all sequence tokens. To further address the limitations of causal attention, EE (Springer et al. 2024) duplicates the input text so that early tokens can attend to subsequent tokens. Although these training-free methods are simple, they still struggle to produce high-quality embeddings consistently. Moreover, approaches such as EE and MetaEOL increase the effective context length, which in turn increases the cost for extracting embeddings.

Training-required Methods To better adapt LLMs for representation learning, a variety of training strategies have been proposed, including supervised contrastive learning (Wang et al. 2024b; Ma et al. 2024; Muennighoff et al. 2024; Li et al. 2025; Wang et al. 2024a; Choi et al. 2024; Muennighoff et al. 2024; Li et al. 2024; BehnamGhader et al. 2024; Su et al. 2025), instruction tuning (Su et al. 2023), and in-context learning (Li et al. 2025). Other research directions explore the use of synthetic data (Wang et al. 2024a; Choi et al. 2024), multi-task learning (e.g., combining generation and representation learning) (Muennighoff et al. 2024; Man et al. 2024), variants of training loss functions (Deng et al. 2025), or the use of pretext tasks (Li et al. 2024; BehnamGhader et al. 2024). Our work falls into the latter category, but focuses on the unique question of whether context compression can be utilized as an effective pretext task to enhance LLMs' ability to learn text representations.

3 Text Representation via Compression

3.1 Method

For LLMs to capture global information from long contexts, we first convert them into bidirectional encoders that can process information in both directions. Building on this, we introduce context compression tasks as pretext tasks designed to further enhance their capacity to model the coherent semantics of entire contexts. Specifically, we consider two context compression tasks, reconstruction and continuation tasks (Ge et al. 2024), as described in detail below.

Reconstruction Task Let g_ϕ denote the original LLM, and f_θ be the targeted LLM-based (bidirectional) encoder adapted from g_ϕ. The encoder f_θ is responsible for producing the embeddings of k special (memory) tokens $\widetilde{m}_1, \ldots, \widetilde{m}_k$ given a long context $n_1, n_2, \ldots, n_a$. We say the set of compressed tokens $\{\widetilde{m}_i\}_{i=1}^k$ effectively captures the context if g_ϕ can reconstruct the context from them. This process is formalized as follows:

$$\widetilde{m_i} = f_\theta(n_1, \ldots, n_a, m_1, \ldots, m_i)[-1] \quad (1)$$

$$\hat{n}_l = g_\phi(\widetilde{m_1} \ldots, \widetilde{m_k}, n_1, \ldots, n_{l-1}) \quad (2)$$

where the memory embedding $\widetilde{m}_i$ for token m_i is taken at the final hidden layer, just before the logit layer, as shown in Equation 1. Equation 2 describes how the original context is reconstructed using the memory embeddings. During training, we sample context $n_1, \ldots, n_a$ from a text collection, and train the encoder f_θ with LoRA (Hu et al. 2022) while keeping the LLM g_ϕ frozen. Here, we use the negative log-likelihood computed by the frozen LLM g_ϕ to compare the original context with the reconstructed context, which is generated from the compressed tokens.

Continuation Task This variant is trained to predict future tokens given a compressed prefix. Formally,

$$\widetilde{m_i} = f_\theta(n_1, \ldots, n_a, m_1, \ldots, m_i)[-1] \quad (3)$$

$$\hat{n}_j = g_\phi(\widetilde{m_1} \ldots, \widetilde{m_k}, n_{a+1}, \ldots, n_{j-1}) \quad (4)$$

where $(n_1, \ldots, n_a, n_{a+1}, \ldots, n_j)$ denotes a sentence from the dataset, split into two segments: a prefix $(n_1, \ldots, n_a)$ and a continuation $(n_{a+1}, \ldots, n_j)$. Similar to the reconstruction task, we adapt the encoder using LoRA and optimize it with the negative log-likelihood (NLL) computed by the frozen LLM g_ϕ, which measures the discrepancy between the generated continuation and the ground-truth continuation.

Continuation Task with Knowledge Distillation (CTKD) To further enhance encoder training, inspired by (Mu, Li, and Goodman 2023; Wingate, Shoeybi, and Sorensen 2022), we propose a third pretext task that combines the continuation objective with knowledge distillation. In this variant, the encoder f_θ is trained not only to generate the correct continuation, but also to match the next-token prediction distribution of the frozen LLM g_ϕ when conditioned on the compressed context versus the original context. This encourages distributional alignment between the two representations. Specifically, we build on the process of Equations 3 and 4, and exploit the Kullback-Leibler divergence loss (KL loss) for training. The loss computation is formalized as follows:

$$\log\left(\frac{g_\phi(n_j \mid n_1, \ldots, n_a, n_{a+1}, \ldots, n_{j-1})}{g_\phi(n_j \mid \widetilde{m_1}, \ldots, \widetilde{m_k}, n_{a+1}, \ldots, n_{j-1})}\right) \quad (5)$$

Mean Pooling Once the LLM has been adapted into the encoder f_θ using one of the proposed pretext tasks, the resulting encoder can be employed to generate text embeddings for a wide range of downstream tasks. In particular, a sentence embedding is obtained by applying mean pooling over the memory embeddings, as follows:

$$z = \frac{1}{k}\sum_{i=1}^{k} \widetilde{m_i}, \quad (6)$$

3.2 Experiment Setup

Implementation Details The unsupervised adaptation of LLMs using the proposed compression-based pretext tasks is conducted on **32,000 samples from the English Wikipedia-103 dataset** (Merity et al. 2017). This choice of dataset for unsupervised adaptation of LLMs to text embedders is consistent with the settings used in LLM2VEC and LLAMA2VEC. This dataset is chosen because Wikipedia is included in the pretraining data of the LLM models, thus the adaptation process does not introduce new factual knowledge; rather, it focuses on teaching the model how to compress contextual information into soft tokens and construct sequence-level representations. For LLM2Comp, we select the default number of memory tokens to be 8, unless stated otherwise. We provide additional details of our training setup and hyperparameters in the Appendix 2.1.

Evaluation Datasets We evaluate the method across 14 diverse tasks in six categories, including clustering, retrieval, semantic textual similarity (STS), classification, and reranking. The semantic similarity tasks (SST) directly assess whether an embedding captures sentence-level semantics. These tasks cover various domains, such as biomedical text, scientific literature, software and programming, finance and banking, and customer support. Dataset details are provided in Table 4 in the Appendix 2.2.

Baseline Methods We evaluate several established methods for sentence representation:

- **LT** and **WMP** (Muennighoff 2022) are training-free methods that obtain embedding from the last token (LT) or weighted mean pooling (WMP).
- **EE (Echo Embedding)** (Springer et al. 2024): Sentence representations are created by duplicating the sentence and applying mean pooling to the latter sentence's tokens.
- **PromptEOL** (Jiang et al. 2024): A prompt, "means in one word:", is appended to the end of the sentence to enhance its representational capacity.
- **MetaEOL** (Lei et al. 2024) designs eight meta-task prompts to form sentence representations.
- **Llama2Vec** (Li et al. 2024) and **LLM2Vec** (BehnamGhader et al. 2024) adapt LLMs for text representation with different pretext tasks, including

Model	Training Samples	Backbone	Clustering			Retrieval			STS			Classification			Reranking		Avg.
			Bior.	Medr.	Twen.	SciF.	NFCo.	Argu.	STS17	SICK-R	STSB.	Bank.	Emot.	Spri.	Stac.	SciD.	
Training-free Methods & Models trained with Pretext Tasks																	
LT.	0	Llama-2	15.99	17.42	15.96	2.17	1.31	14.24	57.8	55.63	45.72	68.65	29.85	47.01	32.07	58.83	33.05
WMP.	0	Llama-2	19.73	19.47	14.54	38.89	6.13	**33.59**	63.91	57.52	58.01	66.42	30.97	58.48	37.74	61.05	40.46
EE.	0	Llama-2	22.94	23.15	25.74	25.61	9.97	25.24	80.51	70.18	71.94	81.79	45.00	68.48	40.79	60.15	46.54
PromptEOL	0	Llama-2	22.49	21.14	31.47	27.16	13.59	11.65	79.67	73.82	75.32	76.37	47.13	26.08	37.65	66.22	43.55
MetaEOL	0	Llama-2	**30.95**	26.56	**40.03**	40.59	**16.41**	21.75	**82.29**	**76.88**	**76.87**	**82.26**	**51.05**	48.24	39.87	77.91	50.83
Llama2vec	32k	Llama-2	22.42	22.25	29.84	16.50	5.22	32.16	75.72	58.00	64.18	75.83	38.64	84.47	28.75	55.51	43.54
LLM2Vec	32k	Llama-2	26.44	25.14	25.76	**44.51**	4.34	31.02	73.45	67.65	65.82	79.77	39.28	70.07	41.48	61.48	46.87
LLM2Comp$_{RC}$	32k	Llama-2	6.65	13.56	8.94	17.41	1.56	14.58	64.66	54.37	41.20	73.95	36.06	76.89	36.50	54.72	35.79
LLM2Comp$_{NLL}$	32k	Llama-2	30.24	**27.34**	37.25	11.93	3.55	24.69	70.65	64.57	63.05	80.12	39.40	72.02	43.36	78.94	46.22
LLM2Comp$_{KL}$	32k	Llama-2	27.79	26.00	31.19	42.57	9.24	30.92	81.56	68.28	70.87	84.33	46.85	**88.81**	**48.20**	**78.18**	**52.49**

Table 1: Performance comparison of different models across three stages of training: self-supervised compression pretraining, evaluated on various tasks from the MTEB benchmark. Each model is assessed on a range of datasets, with results showing the impact of different training approaches on task performance.

EBAE, EBAR, and MNTP. The models are obtained by training LLama2 with pretext tasks using the same dataset as our method (LLM2Comp). Note that here, we consider LLM2Vec and Llama2Vec trained with pretext tasks, without subsequent contrastive learning.

- **LLM2Comp**: We compare three alternatives of our method, including LLM2Comp$_{RC}$ that is based on the reconstruction task, LLM2Comp$_{NLL}$ that is trained with continuation task and NLL loss (CT-NLL), and LLM2Comp$_{KL}$ which is trained with continuation task and knowledge distillation (CTKD).

3.3 Experimental Results

Table 1 presents the performance of training-free methods as well as unsupervised adaptation methods based on different pretext tasks. It is observable that the reconstruction objective offers only marginal gains: LLM2Comp$_{RC}$ performs only slightly better than simple last-token pooling (LT). In contrast, continuation-based objectives lead to better improvements, making LLM2Comp$_{NLL}$ matches the performance of Llama2Vec. However, *the CTKD objective proves to be more suitable than the CT-NLL objective for text representation*, leading to the superior performance of LLM2Comp$_{KL}$ over LLM2Comp$_{NLL}$ and all other baselines.

Stability of Different Compression Tasks In our experiments, we observe that LLM2Comp$_{KL}$ exhibits more stable training behavior compared to LLM2Comp$_{NLL}$ and LLM2Comp$_{RC}$. As shown in Figure 2, LLM2Comp$_{KL}$ achieves a standard deviation of 1.37, which is significantly lower than 2.65 for LLM2Comp$_{RC}$ and 5.32 for LLM2Comp$_{NLL}$. Among the three, LLM2Comp$_{NLL}$ is the most unstable, as its performance can reach 51.85, comparable to LLM2Comp$_{KL}$, but in unfavorable cases, its performance drops to 42.95.

The Impact of Token Length Figure 3 shows the impact of the memory token length on the performance of LLM2Comp$_{KL}$. When the number of tokens is in the range of [1, 8], performance remains stable across most tasks, except for retrieval, which is more sensitive to this hyperparameter. Specifically, when the number of tokens increases to 16, we observe a clear performance drop with the retrieval task. This

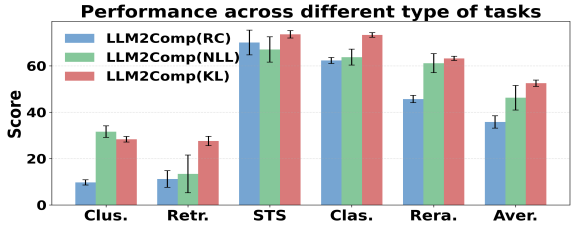


Figure 2: Mean and standard variation of LLM2Comp$_{RC}$, LLM2Comp$_{NLL}$, and LLM2Comp$_{KL}$across different task types, computed over five runs.

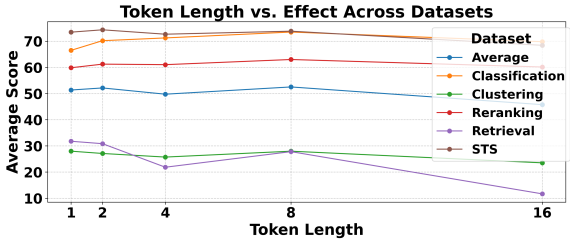


Figure 3: LLM2Comp$_{KL}$: Token length and its Effect

observation contrasts with findings in context compression literature (Qin et al. 2024; Ge et al. 2024; Wingate, Shoeybi, and Sorensen 2022), where using a larger number of tokens (on the order of 100) facilitates downstream generation.

4 Dimensional Collapse

Since compression pretraining aims to condense long contexts into compact memory tokens, it may lead to dimensional collapse, a phenomenon also observed in self-supervised learning (Shwartz-Ziv and LeCun 2024). Here, dimensional collapse occurs when embedding vectors occupy a subspace of significantly lower dimensionality than the original embedding space. In our problem, this manifests in two ways: (i) the pooled embeddings exhibit low rank, and (ii) the resulting memory tokens become highly similar to one another. To study this effect, we analyze embeddings produced by LLM2Comp$_{KL}$ and LLM2Comp$_{NLL}$ over a large corpus of 60K randomly sampled from the SciDocsRR dataset.

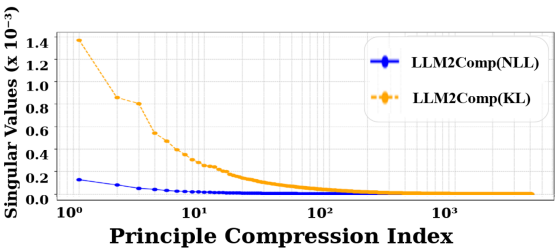


Figure 4: Comparing singular values of LLM2Comp$_{NLL}$ and LLM2Comp$_{KL}$.

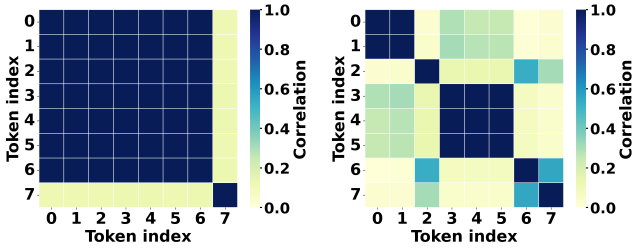


Figure 5: **Llm2Comp**$_{NLL}$ training with 32000 samples (left) and 128000 samples (right) in a bad training case.

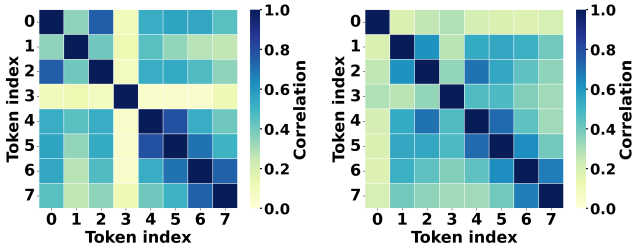


Figure 6: **Llm2Comp**$_{KL}$ training with 32000 samples (left) and 128000 samples (right).

4.1 Effective Dimension After Meanpooling

In dimensional collapse, only a small number of effective dimensions represent the data, with the remaining dimensions being linear combinations of these. The degree of collapse is quantified using Singular Value Decomposition (SVD) (Jing et al. 2022), where dimensions with near-zero singular values are considered ineffective. We first construct an embedding matrix from $N_s = 60{,}000$ samples, each represented by an embedding vector as defined in Equation 6. The covariance matrix of these embeddings is then computed, followed by SVD, and the singular values are sorted in descending order. The sorted values correspond to the principal component index, with the first component having the largest singular value. We plot the singular values against principal component indices for LLM2Comp$_{KL}$ and LLM2Comp$_{NLL}$ in Figure 4. The plot shows that the curve for LLM2Comp$_{NLL}$ drops to zero more rapidly than for LLM2Comp$_{KL}$, with the effective dimensionality for LLM2Comp$_{NLL}$ and LLM2Comp$_{KL}$ being around 10 and 100, respectively. This suggests LLM2Comp$_{NLL}$ experiences more severe dimensional collapse. Intuitively, the KL divergence acts as a regularizer that better preserves information from less frequent tokens, thereby mitigating dimensional collapse. Nevertheless, the effective dimensionality of LLM2Comp$_{KL}$ remains small relative to the total dimension (4096), indicating that there is still room for improvement.

4.2 Effective Token Index Before Meanpooling

The dimensional collapse problem observed in models trained with compression objectives also manifests as a high degree of correlation among the memory tokens. To examine this, we compute the correlation matrix and average it over N_s samples from ScidocRR as follows:

$$Cor = \frac{1}{N_s} \sum_{i=1}^{N_s} \widetilde{M_i} \cdot \widetilde{M_i}^T \quad (7)$$

Here, $\widetilde{M_i}$ denotes the matrix containing the compression tokens $\widetilde{m_{i1}}, \ldots, \widetilde{m_{ik}}$ for the i-th sample. The correlation matrices of a typical LLM2Comp$_{NLL}$ model trained with $N_s = 32$K and $N_s = 128$K are presented in Figure 5. For comparison, Figure 6 shows the corresponding correlation matrices for a representative LLM2Comp$_{KL}$ model trained with the same sample sizes.

Figure 5 shows that tokens from LLM2Comp$_{NLL}$ are highly similar with 32K training samples. In contrast, LLM2Comp$_{KL}$ with 32K samples suffers less from this issue. This observation is consistent with the analysis of effective dimensionality in the previous section. When the number of training samples increases to 128K, the token correlation problem in LLM2Comp$_{NLL}$ is partially alleviated. This improvement is also reflected in its performance, which rises from 42.95 to 48.38 as the sample size increases from 32K to 128K. These results suggest that the degree of token similarity also has a significant impact on the downstream task.

To further verify the above hypothesis, we investigate how reducing correlated tokens impacts downstream performance. We define a *token cluster* as a set of tokens with high pairwise similarity. Initially, each token is treated as its cluster. Clusters are then merged if the minimum similarity between any pair of tokens across clusters exceeds 0.9. From each resulting cluster, we randomly select one representative token, and refer to these selected tokens as *effective tokens*. Sentence embeddings are then computed by applying mean pooling to the embeddings of these effective tokens. Using this procedure, the performance of LLM2Comp$_{NLL}$ trained on 128,000 samples increases from 48.38 to 51.09, as shown in Figure 7. We also include the performance of LLM2Comp$_{KL}$ using a single compression token, as shown in Figure 7. The results indicate that the single-token setting performs unsatisfactorily, likely due to excessive information loss. This highlights the need for a more effective learning strategy that better balances information preservation and redundancy.

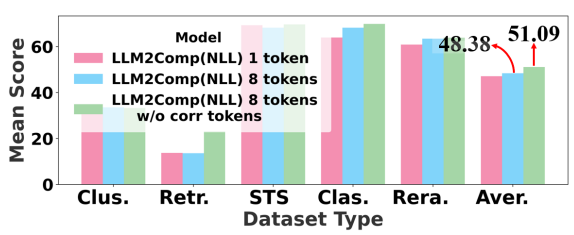


Figure 7: Comparison between LLM2Comp$_{NLL}$ with 1-token compression and 8-token compression (with and without redundant tokens).

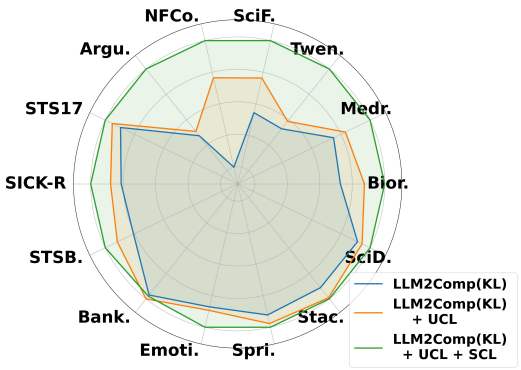


Figure 8: Radar plot showing the performance of LLM2Comp$_{KL}$ across different training stages and datasets.

5 Post-training with Contrastive Learning

A good representation should satisfy two properties (Wang and Isola 2020; Jing et al. 2022): (1) *Alignment*, where semantically related texts are close, and (2) *High Effective Dimension* (Jing et al. 2022), meaning embedding vectors fill the space. However, compression-based objectives often lead to dimensional collapse. Prior work (Jing et al. 2022) shows that contrastive learning mitigates this by pushing negative samples apart. This paper investigates whether post-training with unsupervised contrastive learning (UCL) followed by supervised contrastive learning (SCL) can address dimensional collapse and improve downstream representations.

5.1 Method

Unsupervised Contrastive Learning Following SimCSE (Gao, Yao, and Chen 2021), we construct *positive samples of a particular sentence through dropout, and treat other sentence samples as negative ones*. Formally, the objective is to train the encoder f_θ to maximize the InfoNCE loss:

$$\max_{\theta} \log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2B} \exp(\text{sim}(z_i, z_k)/\tau)} \quad (8)$$

Where z_i and z_j are the representations of the sentences obtained by meanpooling the memory tokens sequences x_i and x_j using f_θ (see Equation 1, and Equation 6). $\text{sim}(\cdot, \cdot)$ denotes the cosine similarity, τ is the temperature hyperparameter, and B is the batch size. For UCL, in-batch negative sampling is used, where the positive embedding of one sample is treated as the negative for others in the batch.

Supervised Contrastive Learning Following UCL, we perform SCL based on supervised data, where relevant pairs are manually annotated. Negative samples are chosen following in-batch negative sampling and hard-negative sampling. The hard-negative samples are pre-chosen by the E5 dataset from a cross-encoder model (Wang et al. 2024b). The optimization objective is similar to UCL (Equation 8) except that we have manually labeled positive samples.

5.2 Experimental Setup

Training Data In the UCL stage, we utilize a Wikipedia sentence subset (128,000 samples) (Gao, Yao, and Chen 2021) identical to LLM2Vec's second-stage training data to ensure comparable experimental conditions. In the SCL stage, we utilize 1,024,000 samples from the public portions of datasets employed in LLM2Vec.

Compared methods In the following, unless otherwise specified, LLM2Comp refers to the model built upon LLM2Comp$_{KL}$ as the foundation model, further enhanced through contrastive post-training. In the UCL stage, we compare LLM2Comp with LLM2Vec, which is also first adapted using a pretext task (MNTP) and then trained with UCL.

In the SCL stage, we further train LLM2Comp initialized from the UCL stage. Our baselines include LLM2Vec, Llama2Vec, and RepLLaMA, all of which share the same LLM backbone and are trained with SCL. RepLLaMA (Ma et al. 2024) directly applies last-token pooling with SCL without using any pretext tasks. Llama2Vec is trained with SCL directly after training with pretext tasks, as reported in the original paper (Li et al. 2024). We also report the performance of several contemporary LLM-based encoders, including ULLME, BGE-ICL (in a zero-shot setting), and Instructor. These models adopt different base models and different training strategies, such as finetuning with in-context data in BGE-ICL (Li et al. 2025), and multi-task learning in ULLME (Man et al. 2024). Consequently, comparisons with these models should be viewed as a reference only, as they do not directly provide scientific insight. Details of the compared methods are given in the Appendix 2.1.

It is worth noting that several of these models, including Instructor, BGE-ICL, and LLM2Vec, have reported results on MTEB, which includes our evaluation datasets. For these models, we directly use the scores reported in their respective papers. For others, such as Llama2Vec and ULLME, which only provide partial MTEB results, we perform our evaluations using their published models.

5.3 Experimental Results

Figure 8 shows the performance of LLM2Comp across different training stages and tasks. The results verify the contribution of UCL and SCL to performance gains beyond pre-training with pretext tasks. In addition, it is observable that the CL stages play a more important role in improving retrieval and clustering tasks.

Compared to other baselines, the experimental results in Table 2 show that LLM2Comp achieves the best performance on most datasets as well as on average. In the UCL

Model	Training Samples	Backbone	Clustering			Retrieval			STS			Classification			Reranking		Avg.
			Bior.	Medr.	Twen.	SciF.	NFCo.	Argu.	STS17	SICK-R	STSB.	Bank.	Emot.	Spri.	Stac.	SciD.	
Unsupervised contrastive learning (UCL)																	
LLM2Vec	160k	Llama-2	31.25	28.04	30.76	64.48	26.81	47.09	86.70	71.77	78.32	84.65	46.58	87.57	47.77	77.62	57.82
LLM2Comp$_{KL}$	160k	Llama-2	32.77	28.32	33.64	59.65	30.91	31.78	87.27	73.69	79.58	86.32	48.56	94.15	51.50	80.94	58.51
LLM2Comp$_{RC}$	160k	Llama-2	7.88	14.97	15.34	52.96	17.16	25.05	76.55	58.52	60.32	75.77	32.55	90.81	42.53	65.39	45.41
LLM2Comp$_{NLL}$	160k	Llama-2	31.03	26.65	35.97	55.49	27.58	27.47	86.61	75.43	77.69	85.85	44.78	91.03	50.95	79.22	56.84
Supervised contrastive learning (SCL)																	
Instructor	1.4M	GTR-XL	30.60	30.80	53.30	64.60	36.00	55.70	90.50	81.70	86.60	82.70	53.20	94.90	52.50	79.50	63.76
ULLME	0.5 M	Phi-1.5	30.46	30.18	42.95	63.41	34.54	55.06	88.49	70.49	80.81	84.24	45.83	92.78	48.61	79.29	60.51
ULLME	0.5 M	Mistral-v0.2	31.48	26.95	38.52	72.86	39.37	45.93	86.38	70.31	78.21	84.57	45.02	92.20	**52.56**	83.47	60.56
ULLME	0.5 M	Llama-3	30.32	26.01	41.32	72.38	39.37	46.78	86.30	69.11	80.25	84.76	49.48	94.73	52.38	81.42	61.05
BGE-ICL	2 M	Mistral-v0.1	35.00	28.10	43.65	**78.10**	40.16	55.81	**91.65**	**83.83**	87.27	87.57	**54.29**	94.79	51.48	84.31	65.43
RepLlama	0.5M	Llama-2	-	-	-	75.60	37.80	45.60	-	-	-	-	-	-	-	-	-
Llama2vec	>3M	Llama-2	30.38	28.21	45.63	75.95	37.38	49.08	66.73	68.57	71.61	77.05	46.17	95.65	45.87	77.04	58.24
LLM2Vec	1.16M	Llama-2	34.81	31.37	51.04	77.30	**40.33**	56.53	90.63	83.01	**88.72**	**88.17**	51.71	**96.83**	51.02	84.03	66.11
LLM2Comp$_{KL}$	0.36M	Llama-2	**37.15**	**33.70**	**55.11**	76.79	39.72	**59.37**	91.40	83.32	86.31	84.57	54.01	96.27	51.90	**85.36**	**66.78**
LLM2Comp$_{RC}$	0.36M	Llama-2	34.81	31.30	53.63	73.54	37.55	57.73	90.90	83.17	86.24	83.61	53.65	95.93	51.91	82.87	65.49
LLM2Comp$_{NLL}$	0.36M	Llama-2	36.53	32.85	53.14	75.05	39.13	58.20	91.61	83.05	85.33	82.99	52.24	96.16	51.45	84.85	65.90

Table 2: Performance comparison of different models across different post-training stages. Here, the backbone models include: Llama-2 (7B), Mistral-v0.1 (7B), GTR-XL (1.5B), Phi-1.5 (1.3B), Mistral-v0.2 (7B), and Llama-3 (8B).

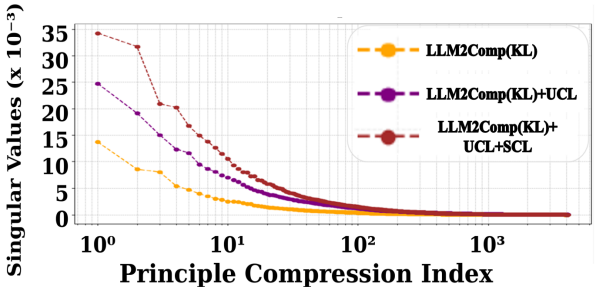


Figure 9: Comparing singular values of LLM2Comp$_{KL}$ in different training stages.

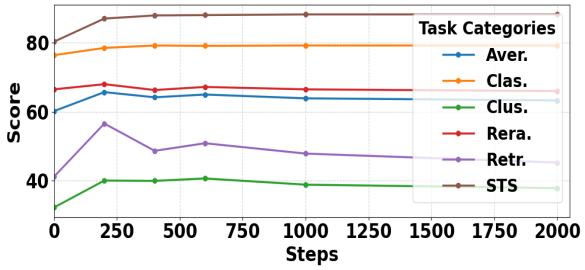


Figure 10: Performance across different training steps.

stage, LLM2Comp surpasses LLM2Vec, confirming that the benefits of our compression-based pretext task carry over to the subsequent training. A similar pattern is observed in the SCL stage after UCL, where LLM2Comp is superior to LLM2Vec, and other contemporary models. Notably, LLM2Comp achieves this using a much smaller amount of supervised data compared to LLM2Vec as shown in Table 2. This suggests that compression-based pretraining provides a stronger foundation, enabling more efficient post-training. Since training cost scales with the amount of supervised data, this also highlights the practical value of LLM2Comp.

5.4 Further Analysis

Contrastive Learning Alleviates Dimensional Collapse As shown in Figure 9, the model trained with LLM2Comp$_{KL}$ + UCL + SCL exhibits a higher effective dimension than the model trained with LLM2Comp$_{KL}$ + UCL. Furthermore, LLM2Comp$_{KL}$ + UCL achieves a higher effective dimension than LLM2Comp$_{KL}$ alone. The reduction of dimensional collapse also correlates with the enhancement of LLM2Comp over different training stages as shown in the previous section.

Convergence Analysis Models trained with the compression objective exhibit higher data efficiency, achieving peak performance with only 0.36M training samples, compared to the 1.16M samples required by LLM2Vec, as shown in Table 2. For a more detailed analysis, Figure 10 shows how the performance of LLM2Comp evolves over training steps in the SCL stage. We observe that our model converges rapidly, reaching optimal performance within 200 steps and remaining stable for most tasks. However, for retrieval tasks, performance begins to decline when contrastive learning continues beyond this point.

6 Conclusion

Our study demonstrates the potential of context compression as a pretext task for the unsupervised adaptation of large language models (LLMs). We identify CTKD as the optimal training objective and determine the appropriate number of memory tokens needed for downstream representations. A deeper analysis shows that CTKD effectively mitigates dimensional collapse, resulting in stronger text representations than other pretext tasks. Building on this, additional contrastive learning yields a robust embedding model, LLM2Comp, which outperforms contemporary baselines (LLM2Vec and Llama2Vec) trained with similar recipes but requires much less training data. Furthermore, we provide insights into the effective dimensionality, task-aware performance, and sample efficiency, highlighting promising directions for future research.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. W2532049).

References

BehnamGhader, P.; Adlakha, V.; Mosbach, M.; Bahdanau, D.; Chapados, N.; and Reddy, S. 2024. LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders. *CoRR*, abs/2404.05961.

Cheng, X.; Wang, X.; Zhang, X.; Ge, T.; et al. 2024. xRAG: Extreme Context Compression for Retrieval-augmented Generation with One Token. In *Annual Conference on Neural Information Processing Systems*.

Chevalier, A.; Wettig, A.; Ajith, A.; and Chen, D. 2023. Adapting Language Models to Compress Contexts. In *Conference on Empirical Methods in Natural Language Processing*.

Choi, C.; Kim, J.; Lee, S.; et al. 2024. Linq-Embed-Mistral Technical Report. *CoRR*, abs/2412.03223.

Deng, J.; Jiang, Z.; Pang, L.; et al. 2025. Following the Autoregressive Nature of LLM Embeddings via Compression and Alignment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Fan, A.; Jernite, Y.; Perez, E.; Grangier, D.; Weston, J.; and Auli, M. 2019. ELI5: Long Form Question Answering. In *Conference of the Association for Computational Linguistics*.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Conference on Empirical Methods in Natural Language Processing*.

Ge, T.; Hu, J.; Wang, L.; Wang, X.; Chen, S.; and Wei, F. 2024. In-context Autoencoder for Context Compression in a Large Language Model. In *The Twelfth International Conference on Learning Representations*.

He, W.; Liu, K.; Liu, J.; et al. 2018. DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*.

Hu, E. J.; Shen, Y.; Wallis, P.; et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations*.

Jiang, T.; Huang, S.; Luan, Z.; Wang, D.; and Zhuang, F. 2024. Scaling Sentence Embeddings with Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP*.

Jing, L.; Vincent, P.; LeCun, Y.; and Tian, Y. 2022. Understanding Dimensional Collapse in Contrastive Self-supervised Learning. In *The Tenth International Conference on Learning Representations*.

Joshi, M.; Choi, E.; Weld, D.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Barzilay, R.; and Kan, M.-Y., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1601–1611. Vancouver, Canada: Association for Computational Linguistics.

Karpukhin, V.; Oguz, B.; Min, S.; et al. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Conference on Empirical Methods in Natural Language Processing*.

Lei, Y.; Wu, D.; Zhou, T.; Shen, T.; Cao, Y.; Tao, C.; and Yates, A. 2024. Meta-Task Prompting Elicits Embeddings from Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Li, C.; Liu, Z.; Xiao, S.; Shao, Y.; and Lian, D. 2024. Llama2Vec: Unsupervised Adaptation of Large Language Models for Dense Retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Li, C.; Qin, M.; Xiao, S.; Chen, J.; et al. 2025. Making Text Embedders Few-Shot Learners. In *The Thirteenth International Conference on Learning Representations*.

Ma, X.; Wang, L.; Yang, N.; Wei, F.; and Lin, J. 2024. Fine-Tuning LLaMA for Multi-Stage Text Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Man, H.; Ngo, N. T.; Dernoncourt, F.; and Nguyen, T. H. 2024. ULLME: A Unified Framework for Large Language Model Embeddings with Generation-Augmented Learning. *CoRR*, abs/2408.03402.

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2017. Pointer Sentinel Mixture Models. In *5th International Conference on Learning Representations*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations*.

Mu, J.; Li, X.; and Goodman, N. D. 2023. Learning to Compress Prompts with Gist Tokens. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems*.

Muennighoff, N. 2022. SGPT: GPT Sentence Embeddings for Semantic Search. *CoRR*, abs/2202.08904.

Muennighoff, N.; Su, H.; Wang, L.; Yang, N.; Wei, F.; Yu, T.; Singh, A.; and Kiela, D. 2024. Generative Representational Instruction Tuning. *CoRR*, abs/2402.09906.

Muennighoff, N.; Tazi, N.; Magne, L.; and Reimers, N. 2023. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; et al. 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*.

Ni, J.; Qu, C.; Lu, J.; Dai, Z.; Ábrego, G. H.; Ma, J.; Zhao, V. Y.; Luan, Y.; Hall, K. B.; Chang, M.; and Yang, Y. 2022. Large Dual Encoders Are Generalizable Retrievers. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 9844–9855. Association for Computational Linguistics.

Nie, Z.; Feng, Z.; Li, M.; Zhang, C.; Zhang, Y.; Long, D.; and Zhang, R. 2024. When Text Embedding Meets Large Language Model: A Comprehensive Survey. *CoRR*, abs/2412.09165.

Qin, G.; Rosset, C.; Chau, E. C.; Rao, N.; and Durme, B. V. 2024. Dodo: Dynamic Contextual Compression for Decoder-only LMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.*, 21: 140:1–140:67.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Su, J.; Duh, K.; and Carreras, X., eds., *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392. Austin, Texas: Association for Computational Linguistics.

Shao, N.; Xiao, S.; Liu, Z.; and Zhang, P. 2024. Flexibly Scaling Large Language Models Contexts Through Extensible Tokenization. *CoRR*, abs/2401.07793.

Shwartz-Ziv, R.; and LeCun, Y. 2024. To Compress or Not to Compress - Self-Supervised Learning and Information Theory: A Review. *Entropy*, 26(3): 252.

Springer, J. M.; Kotha, S.; Fried, D.; Neubig, G.; and Raghunathan, A. 2024. Repetition Improves Language Model Embeddings. *CoRR*, abs/2402.15449.

Su, C.; Shi, D.; Huang, S.; et al. 2025. Training LLMs to be Better Text Embedders through Bidirectional Reconstruction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Su, H.; Shi, W.; Kasai, J.; Wang, Y.; et al. 2023. One Embedder, Any Task: Instruction-Finetuned Text Embeddings. In *Findings of the Association for Computational Linguistics*.

Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; and Mittal, A. 2018. FEVER: A Large-scale Dataset for Fact Extraction and VERification. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Wang, L.; Yang, N.; Huang, X.; Yang, L.; Majumder, R.; and Wei, F. 2024a. Improving Text Embeddings with Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Wang, L.; Yang, N.; Huang, X.; Yang, L.; Majumder, R.; and Wei, F. 2024b. Multilingual E5 Text Embeddings: A Technical Report. *CoRR*.

Wang, T.; and Isola, P. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*.

Wingate, D.; Shoeybi, M.; and Sorensen, T. 2022. Prompt Compression and Contrastive Conditioning for Controllability and Toxicity Reduction in Language Models. In *Findings of the Association for Computational Linguistics: EMNLP*.

Xie, X.; Dong, Q.; Wang, B.; Lv, F.; Yao, T.; Gan, W.; Wu, Z.; Li, X.; Li, H.; Liu, Y.; and Ma, J. 2023. T2Ranking: A Large-scale Chinese Benchmark for Passage Ranking. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Xu, Y.; Feng, Y.; Mu, H.; Hou, Y.; et al. 2024. Concise and Precise Context Compression for Tool-Using Language Models. In *Findings of the Association for Computational Linguistics*.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; et al. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Zhang, X.; Ma, X.; Shi, P.; and Lin, J. 2021. Mr. TyDi: A Multi-lingual Benchmark for Dense Retrieval. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*.

Zhang, X.; Thakur, N.; Ogundepo, O.; Kamalloo, E.; Alfonso-Hermelo, D.; Li, X.; Liu, Q.; Rezagholizadeh, M.; and Lin, J. 2023. MIRACL: A Multilingual Retrieval Dataset Covering 18 Diverse Languages. *Transactions of the Association for Computational Linguistics*, 11: 1114–1131.