Decoder-Only LLMs can be Masked Auto-Encoders

Dan Qiao[1,2], **Yuan Gao**[2] **Zheming Yang**[2], **Di Yang**[2],
Ziheng Wu[2], **Pengcheng Lu**[2], **Minghui Qiu**[2], **Juntao Li**[1*] **Min Zhang**[1]
[1]School of Computer Science and Technology, Soochow University
[2]ByteDance Inc
qiaodan.23@bytedance.com

Abstract

Modern NLP workflows (e.g., RAG systems) require different models for generation and embedding tasks, where bidirectional pre-trained encoders and decoder-only Large Language Models (LLMs) dominate respective tasks. Structural differences between models result in extra development costs and limit knowledge sharing between tasks. In this work, we present UniMAE, a novel unsupervised training method that transforms a Decoder-Only LLM into a **Uni**-Directional **M**asked **A**uto-**E**ncoder. UniMAE compresses high-quality semantic information into the [EOS] embedding while preserving the generation capabilities of LLMs. Comprehensive evaluations across 56 MTEB datasets demonstrate that UniMAE can achieve state-of-the-art results under unsupervised settings with merely 100 training steps, establishing the first effective approach to unifying generation and representation learning in decoder-only architectures.

1 Introduction

Pre-trained language models have been widely applied in various scenarios (Devlin, 2018; Ouyang et al., 2022; Brown et al., 2020; Wu et al., 2025; **?**). However, the workflows of real-world applications like RAG (Fan et al., 2024) or retrieval systems (Zhu et al., 2023) often involve collaboration between generation tasks and embedding tasks. For example, RAG requires retrieval followed by generation (Asai et al., 2023), while retrieval systems often need to rewrite queries before performing the search (Ma et al., 2023; Liu and Mozafari, 2024).

For embedding tasks, encoder models with bidirectional attention mechanisms have been the mainstream choice (Devlin, 2018; Liu, 2019). More recently, decoder-only large language models (LLMs) with unidirectional attention have demonstrated the ability to compress knowledge from trillions of tokens during pre-training (Radford, 2018; Ouyang et al., 2022; Guo et al., 2025), thus have been widely utilized in various generative tasks. This structural separation leads to extra training and deployment costs and impedes knowledge sharing between tasks (Asai et al., 2023). In the long run, enabling large language models (LLMs) to perform both generation and embedding tasks simultaneously holds significant promise.

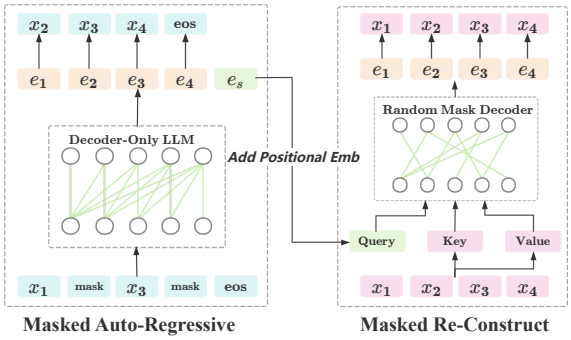


Figure 1: Overview of UniMAE training. The left part shows the Masked Auto-Regressive process, while the right part shows the reconstruction process using a tiny decoder with [EOS] embeddings to recover the input.

In this paper, we introduce UniMAE, an unsupervised method that compresses the semantic content of input text into the [EOS] (end-of-sequence) embedding of the last transformer layer, while preserving the capabilities of original LLMs (Figure 1). We scale our method across models with 1B, 3B, and 8B parameters and conduct evaluations on 56 datasets of the Massive Text Embeddings Benchmark (MTEB) (Muennighoff et al., 2022). UniMAE achieves state-of-the-art performance on MTEB under an unsupervised setting within only 100 training steps. Evaluations on language modeling tasks demonstrate that UniMAE not only enhances representation capabilities but also maintains the generation abilities of LLMs. Domain post-training experiments indicate that UniMAE can simultaneously improve the generation and embedding performance in a single training process.

* Corresponding author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 713–723

To our knowledge, UniMAE is the first effective approach to unifying generation and representation learning in decoder-only architectures.

2 Method

2.1 Motivation

Unlike pre-trained encoder models that utilize special tokens like [CLS] for sentence embeddings, LLMs primarily focus on predicting the next token during pre-training and lack a dedicated position for sentence representation. As a result, neither averaging the token embeddings nor directly using the [EOS] embedding can effectively represent the entire sentence. (Springer et al., 2024; Jiang et al., 2023; BehnamGhader et al., 2024).

To bridge this gap, recent studies have focused on enhancing the representational capabilities of large models. Many directly use the model as an encoder, employing average embeddings or token-level embeddings for supervised contrastive fine-tuning (Lee et al., 2024; Li et al., 2024). Springer et al. (2024) and BehnamGhader et al. (2024) argue that unidirectional attention prevents each token from incorporating information from subsequent tokens, thereby limiting the model's contextual understanding. Springer et al. (2024) addresses this limitation by repeating the sentence twice. Meanwhile, BehnamGhader et al. (2024) applies a Bidirectional Attention Mechanism and adapts the MLM task to LLM, then finally uses average pooling to get sentence embeddings.

However, these methods still fail to address the lack of a unified pooling mechanism in LLMs. Moreover, they simply repurpose LLMs as encoders during fine-tuning, thereby overlooking the potential loss of their generative capabilities.

2.2 Overall framework

Similar to Visual-MAE (He et al., 2022), UniMAE employs the [EOS] embedding as the latent representation of the corrupted input and initializes a small decoder to reconstruct the original input based on this representation. Unlike other approaches, our method maintains a unidirectional attention mechanism characteristic of large language models (LLMs). We illustrate the entire training process in Figure 1.

The overall objective consists of two components: 1. Masked Auto-Regressive (MAR) and 2. Masked Re-Construct (MRC). The complete training process involves jointly optimizing two sets of parameters: the LLM parameters Φ_{llm} and small decoder parameters Φ_{dec}.

2.3 Masked Auto-Regressive (MAR)

The sole distinction between the classic Auto-Regressive (Radford, 2018; Radford et al., 2019; Brown et al., 2020) and MAR is that the input is corrupted with random masks of ratio p_{mar}. For a sentence X, it is corrupted by replacing some tokens with mask tokens, resulting in the masked sentence $\tilde{X}$. The MAR loss can be expressed as:

$$L_{MAR} = \sum_{t} CE(x_t \mid \tilde{x}_{i<t}; \Phi_{llm}), \quad (1)$$

where $x_i \in X$ is the i-th token or the original text, $\tilde{x}_i \in \tilde{X}$ is the i-th token of the corrupted text. Φ_{llm} denotes the parameters of the original LLM.

2.4 Masked Re-Construct (MRC)

We believe that the representation at [EOS] position has the potential to encapsulate the semantic meaning of the entire input. Since only the last token can attend to all the input tokens under the unidirectional attention mechanism. So the latent representation h_{eos} is:

$$h_{eos} \leftarrow \Phi_{llm}\left(\tilde{X}\right) \quad (2)$$

The Masked Re-Construct (MRC) process utilizes a newly initialized small decoder, which employs the [EOS] embedding combined with partial input information to reconstruct the original sentence. The input of the tiny decoder can be formulated as:

$$\begin{aligned} H_1 &\leftarrow [h_{eos} + p_0, \ldots, h_{eos} + p_N], \\ H_2 &\leftarrow [h_{eos}, e_{x_1} + p_1, \ldots, e_{x_N} + p_N], \end{aligned} \quad (3)$$

where p_i is the trainable positional embedding, e_i is the input token embedding. The only difference between MRC and traditional cross-attention is that we applied a mask to the attention matrix, forcing the model to rely more on the h_{eos} for sentence recovery. We set a mask ratio p_{mrc} and let $B_{ij} \sim \text{Bernoulli}(p_{mrc}) \in \{0, 1\}$. Then the computation process of attention can be expressed with the following formula:

$$\begin{aligned} Q = H_1 W^Q, K = H_2 W^K, V = H_2 W^V \\ M_{ij} = \begin{cases} 0, & \text{if } i \neq j \wedge B_{ij} = 1 \\ -\infty, & \text{else} \end{cases}, \\ A = \text{softmax}\left(\frac{Q^T K}{\sqrt{d}} + M\right) V. \end{aligned} \quad (4)$$

Attn.	Categories → # of datasets →	Retr. 15	Rerank. 4	Clust. 11	PairClass. 3	Class. 12	STS 10	Summ. 1	Avg 56
				LLaMA3.2-1B					
Uni.	MEAN	15.13	42.96	34.30	53.50	56.20	56.27	28.06	39.32
Uni.	ECHO	24.57	48.27	36.68	67.65	65.60	71.86	29.79	48.28
Bi.	MNTP	18.53	43.54	34.20	56.30	60.24	61.15	27.21	42.12
Uni.	UniMAE	**34.48**	50.51	38.42	75.91	68.88	73.14	29.93	52.81
Bi.	MNTP + SimCSE	30.59	49.67	37.48	74.41	65.70	72.85	**30.71**	50.73
Uni.	UniMAE + SimCSE	31.54	**54.23**	**43.22**	**78.04**	**69.26**	**76.95**	30.14	**54.11**
				LLaMA3.2-3B					
Uni.	MEAN	14.61	43.22	34.73	55.50	55.55	56.00	22.13	39.09
Uni.	ECHO	25.44	48.64	36.86	70.68	67.96	71.95	24.31	49.16
Bi.	MNTP	17.75	44.69	36.06	57.47	59.79	61.87	23.06	42.38
Uni.	UniMAE	34.25	51.79	38.95	78.31	70.14	73.06	29.25	53.30
Bi.	MNTP + SimCSE	34.35	50.91	39.83	76.21	66.00	73.77	**30.64**	52.61
Uni.	UniMAE + SimCSE	**36.09**	**55.41**	**43.77**	**80.90**	**71.60**	**76.53**	30.33	**56.11**
				LLaMA3.1-8B					
Uni.	MEAN	14.26	46.89	37.84	70.12	66.73	62.32	24.98	42.95
Uni.	ECHO	25.53	49.75	37.77	70.11	67.03	71.83	27.32	49.24
Bi.	MNTP	22.83	48.46	40.96	64.26	60.11	64.50	27.40	45.95
Uni.	UniMAE	35.37	52.74	41.25	78.23	68.49	73.45	30.40	53.87
Bi.	MNTP + SimCSE	**39.75**	53.59	42.70	78.71	67.38	**76.84**	**31.77**	55.81
Uni.	UniMAE + SimCSE	38.51	**56.91**	**46.06**	**80.15**	**68.77**	76.21	29.80	**56.60**

Table 1: Results on 56 MTEB datasets with best results highlighted in bold, and the second-best results underlined.

The MRC object can be formalized as:

$$\mathcal{L}_{MRC} = \sum_{t} CE(x_t \mid A, H_1, H_2, \Phi_{dec}). \quad (5)$$

The joint optimization of MAR and MRC is expressed as follows:

$$\mathcal{L}_{UniMAE} = \alpha \mathcal{L}_{MAR} + \mathcal{L}_{MRC}, \quad (6)$$

where α is the weight to control the MAR objective.

3 Experiment

3.1 Settings

For embedding tasks, we conduct evaluations on the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2022), a collection of 7 diverse embedding task categories covering a total of 56 datasets. All evaluations are done using the official MTEB code repository. For preliminary experiments and ablations, we use MTEB-15 a subset including 15 representative tasks. Details are attached in Appendix A.

For baselines, we compare our method with average pooling, Echo (Springer et al., 2024), which repeats the sentence twice and averages the token embedding of the second sentence, and LLM2Vec (BehnamGhader et al., 2024), which sequentially employs MNTP and SimCSE (Gao et al., 2021b) to train the model. To make a comparison with LLM2Vec with SimCSE training, we also provide UniMAE with further SimCSE training. We provide training details in Appendix C.

3.2 Embedding Performance

Table 1 shows that, after UniMAE + SimCSE training, we observe a considerable improvement in performance for all three models compared with direct mean pooling, which is 28% for the 1B model, 43% for the 3B model and 32% for the 8B model. Using only UniMAE can significantly surpass all single unsupervised learning methods, UniMAE even outperforms MNTP + SimCSE training on 1B and 3B models. Overall, the UniMAE framework achieves the state-of-the-art results across models of all scales. Scores for each dataset are shown in Appendix G.

3.3 Will UniMAE hurt LLMs?

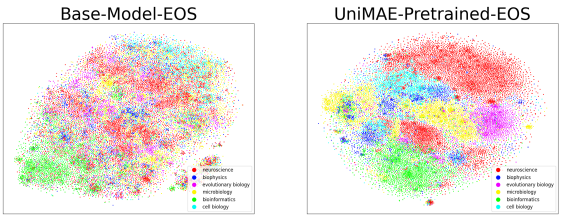
We selected 10 commonly used datasets for evaluating generative language models, including ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019) and MMLU (Hendrycks et al., 2021). Details can be found in Appendix E. As shown in Figure 3b, after UniMAE training, the generative capabilities of the model are nearly equivalent to those of the original foundation models. We also tested the baseline method, MNTP training, which modifies causal attention to bidirectional attention. We observe that the structural modification significantly

reduces the model's performance on language modeling tasks. The same situation occurs in supervised / unsupervised contrastive learning applied to the representations after average pooling.

3.4 UniMAE optimizes Vector Space

We apply t-SNE transformations to the top 6 classes' sentence embeddings from BiorxivClusteringS2S dataset, using PCA initialization, 2000 iterations, and early exaggeration of 20.

Figure 2 shows the distribution of the UniMAE and baseline embeddings. UniMAE improves spatial distribution by bringing samples closer within each group using [EOS] token embeddings. This improved distribution highlights the effectiveness of our approach in boosting LLM performance as an encoder.



(a) Base Model (b) UniMAE

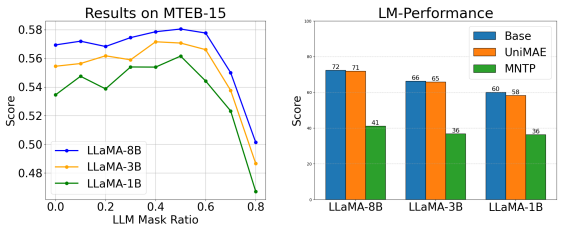
Figure 2: t-SNE visualization of sentence embeddings from top 6 classes on BiorxivClusteringS2S testset.

3.5 Is MAR necessary?

We designed MAR to prevent the model from learning the shortcut of simply memorizing all the input tokens during training. Therefore, we introduce noise to the input, allowing the model to extract semantic representations even from incomplete inputs, rather than simply memorizing them. To prove that MAR is superior to AR, we trained different models using input masking ranging from 0% to 80% and performed inference on MTEB-15. As shown in Figure 3a, for models of different sizes, the optimal mask ratio falls between 40% and 60%. For convenience, we set this ratio to 50% for all models. In the detailed results, we observe that input masking significantly improved performance on tasks related to semantic similarity, such as Clustering and STS, while the enhancement for retrieval tasks was relatively minor.

3.6 Domain Post-Training Result

We believe that UniMAE can not only enhance the representation ability of LLMs but also serve as a novel pre-training and post-training method to simultaneously improve both generative and representational capabilities. We conduct E-Commerce domain post-training for 100B tokens, we describe more details in the Appendix F. The evaluation contains three types of tasks: domain language modeling, retrieval, and classification. Language model tasks (LM.) are primarily QA, retrieval (Retr.), and classification (Class.) tasks that are similar to those in MTEB. Table 2 demonstrates that UniMAE training slightly outperforms Auto-Regressive in generation tasks, and significantly outperforms Auto-Regressive in embedding tasks.



(a) Mask Ratio Search (b) LM Evaluation

Figure 3: (a) Results under different MAR mask ratio on MTEB-15. (b) Performance on language model tasks.

Categories → **# of sub-tasks** →	**LM.** 3	**Retr.** 7	**Class.** 7
Qwen2-0.5B	71.12	36.91	81.43
Qwen2-0.5B + AR	73.16	35.89	83.98
Qwen2-0.5B + UniMAE	**74.55**	**50.84**	**95.38**

Table 2: Model performance after domain post-training

4 Conclusion

We introduce UniMAE, a method capable of training models for both generation and embedding tasks. We evaluate UniMAE on diverse datasets, including open-source and domain-specific corpora, across both types of tasks. UniMAE achieves state-of-the-art performance on the MTEB benchmark without compromising the language modeling capabilities of LLMs. As a domain post-training approach, UniMAE enhances both the representational and generative capacities of models simultaneously. In future work, we plan to apply UniMAE for large-scale pre-training from scratch and instruction alignment, aiming to develop a new generation of LLMs that excel in both generation and embedding tasks.

5 Limitations

There is an inherent limitation: we initialized a small decoder from scratch to reconstruct the original input. However, during the inference process, we directly discarded this small decoder and instead utilized the results generated by the LLM for both generation and representation tasks. This approach leads to a certain degree of parameter waste. If this method is scaled up for large-scale zero pretraining, such waste would be suboptimal. In the future, we will explore new structural designs to address this issue.

Acknowledgments

We want to thank all the anonymous reviewers for their valuable comments. This work was supported by the National Science Foundation of China (NSFC No. 62206194), the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20220488), the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001), and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

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

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021a. A framework for few-shot language model evaluation.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. 2017. Dbpedia-entity v2: A test collection for entity search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 1265–1268. ACM.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2023. Scaling sentence embeddings with large language models. *arXiv preprint arXiv:2307.16645*.

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. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Chaofan Li, MingHao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Yingxia Shao, Defu Lian, and Zheng Liu. 2024. Making text embedders few-shot learners. *arXiv preprint arXiv:2409.15700*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3622–3628.

Jie Liu and Barzan Mozafari. 2024. Query rewriting via large language models. *arXiv preprint arXiv:2403.09060*.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *arXiv preprint arXiv:2305.14283*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

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.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2024. Repetition improves language model embeddings. *arXiv preprint arXiv:2402.15449*.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106.

Ziheng Wu, Zhenghao Chen, Ruipu Luo, Can Zhang, Yuan Gao, Zhentao He, Xian Wang, Haoran Lin, and Minghui Qiu. 2025. Valley2: Exploring multimodal models with scalable vision-language design. *arXiv preprint arXiv:2501.05901*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.

A Massive Text Embeddings Benchmark (MTEB)

The original MTEB benchmark (Muennighoff et al., 2022) consists of 56 datasets of various embedding tasks. Some tasks like classification and clustering require additional use of small clustering models or classification heads after extracting embeddings. For a fair comparison, we use the official MTEB code repository [1] to test all the models, simply overriding the encode function to extract representations for the given text. Since the models under evaluation have different architectures and pooling methods, they may have different optimal instructions for their respective tasks. For fair comparisons, we do not use any instruction and only extract embeddings from the test texts.

MTEB consists of diverse small and large embedding tasks. Some datasets like MSMARCO (Nguyen et al., 2016) and DBPedia (Hasibi et al., 2017) have even more than 5,000,000 samples. Considering the test speed, for preliminary experiments and ablation studies, we select 15 representative tasks as a subset in Table 3, which aligns with LLM2VEC (BehnamGhader et al., 2024). For each

[1] https://github.com/embeddings-benchmark/mteb

task category, we selected the same proportion of datasets to ensure that the results in MTEB-15 are not biased towards MTEB-56.

Category	Dataset
Retrieval (3)	SciFact ArguAna NFCorpus
Reranking (2)	StackOverflowDupQuestions SciDocsRR
Clustering (3)	BiorxivClusteringS2S MedrxivClusteringS2S TwentyNewsgroupsClustering
Pair Classification (1)	SprintDuplicateQuestions
Classification (3)	Banking77Classification EmotionClassification MassiveIntentClassification
STS (3)	STS17 SICK-R STSBenchmark
Overall	15 datasets

Table 3: Datasets of MTEB-15

B Usage Instructions

In the entire framework, there are two important hyper-parameters: the mask ratio p_{mar} of MAR, and the weight α of MAR. Based on our experience, a higher p_{mar} often provides greater benefits for tasks such as STS and clustering, which are sensitive to vector space comparisons, while the improvement for retrieval tasks is relatively smaller. For the weight $alpha$, it depends on your expectations for the LLM base. If you aim for the LLM itself to learn from the data (wiki-text), you can set a relatively high α. However, if your goal is to enhance the representation capability while maintaining the original abilities of the LLM, a value of 0.1 would be sufficient.

Regarding the number of training steps, we only trained for 100 steps on general open-source data. This is mainly because LLMs have typically encountered this open-source data during their pretraining. Numerous studies have shown that performing autoregressive training on repetitive data can easily lead to model collapse. Therefore, if there is data that can enhance the LLM itself, it can be trained extensively, and full-parameter training can be employed. For instance, we conducted full-parameter training on domain-specific data with 100 billion tokens, which continuously improved the model's overall capabilities. However, if training on data does not benefit LLMs or even hurt LLMs, we recommend using LoRA to converge new parameters instead.

C Unsupervised Training Settings

C.1 Baselines

For Echo (Springer et al., 2024), we simply repeat the input text and take the average output embedding of the second sentence as the global embedding. For MNTP and MNTP + SimCSE (BehnamGhader et al., 2024), we train the LLaMA-3.2-1B and LLaMA-3.2-3B using the official code with the setting mentioned for Sheared-LLaMA-1.3B, as they do not release models based on these two foundation models. For LLaMA-3.1-8B, we directly use the released lora weights [2].

C.2 Our method

We keep the same training settings across all the models. For UniMAE training, we train all the models using Peft [3] package with LORA (Hu et al., 2021) adapter to train the models, with $\alpha_{lora} = 32, r = 16, lr = 1e-4$. We only add trainable lora weight to the attention block and mlp block. We train the model on randomly selected wiki-text data [4] for 100 steps with a global batch size of 32 samples. The max length of each sample is 512. The learning rate is $1e-4$ and we use a constant learning rate scheduler. The MAE weight α is set to 0.1 and MRC weight β is set to 1. MAR mask ratio p_{mar} is set to 0.5, and MRC mask ratio p_{mrc} is set to 0.5.

For SimCSE training, we train all the models using Peft package with LORA (Hu et al., 2021) adapter to train the models, with $\alpha = 32, r = 16, lr = 1e-4$. We only add trainable lora weight to the attention block and mlp block. We train the model on randomly selected wiki-data for 100 steps with a global batch size of 64 samples. The max length of each sample is 512. The learning rate is $1e-4$ and we use a constant learning rate scheduler.

D Combine UniMAE with Supervised Contrastive Learning

Experiments in table 1 have shown that UniMAE significantly improves the embedding performance

[2] https://huggingface.co/McGill-NLP/LLM2Vec-Meta-Llama-3-8B-Instruct-mntp

[3] https://github.com/huggingface/peft

[4] https://huggingface.co/datasets/wikimedia/wikipedia

Categories →	Retr.	Rerank.	Clust.	PairClass.	Class.	STS	Avg
# of datasets →	3	2	3	1	3	3	15
LLaMA3.2-1B							
Base Model	51.17	66.08	37.23	93.76	69.70	81.85	63.05
UniMAE + SimCSE	**53.95**	**67.17**	**38.45**	**94.57**	**70.41**	**84.00**	**64.62**
LLaMA3.2-3B							
Base Model	50.77	66.18	**40.69**	88.89	67.53	79.81	62.51
UniMAE + SimCSE	**56.24**	**69.32**	40.23	**94.25**	**70.28**	**83.50**	**65.58**
LLaMA3.1-8B							
Base Model	53.07	67.19	40.21	95.04	69.42	82.44	64.32
UniMAE + SimCSE	**57.93**	**70.67**	**42.65**	**96.13**	**70.77**	**84.51**	**67.00**

Table 4: Results on MTEB-15 after supervised contrastive learning.

of LLMs. Many studies have proven that fine-tining supervised contrastive learning data can further improve the performance of LLMs on embedding tasks. To see whether our method can further improve the performance under supervised settings, we collect open-sourced supervised contrastive learning data to train the base model and model trained using UniMAE + SimCSE. We follow the setting of BGE-ICL (Li et al., 2024), which also does not modify the model architecture, using [EOS] embedding as the sentence embedding for contrastive learning. We use the official training code of BGE-ICL [5]. Data can be found here [6]. For base models and models trained using UniMAE + SimCSE, we both use the [EOS] embedding of the last transformer layer as the sentence embedding to conduct supervised contrastive learning. We train each model for 4000 steps with a global batch size of 64 samples. The max length of each text is set to 512. We use the Peft package with LORA (Hu et al., 2021) adapter to train the models. The learning rate is $1e-4$ and we use a constant learning rate scheduler.

The result can be found in table 4. It demonstrates that our approach not only directly enhances the base model's capabilities but also shows improved performance after additional contrastive learning compared to the original base model.

E Language Model Evaluation

We use the lm-evaluation-harness package (Gao et al., 2021a) to evaluate the LLMs on 10 downstream tasks: ARC-E (Clark et al., 2018), LAMBADA (Paperno et al., 2016), LogiQA (Liu et al., 2020), PIQA (Bisk et al., 2020), SciQ (Welbl et al., 2017), and WinoGrande (Sakaguchi et al., 2021), HellaSwag (Zellers et al., 2019), ARC-C (Clark et al., 2018)), Natural Questions (Kwiatkowski et al., 2019), MMLU (Hendrycks et al., 2021). We keep a zero-shot setting for each task.

F E-commerce Domain Post-training

In the E-commerce domain, the pre-training dataset consists of approximately 100 billion tokens (Due to the principle of anonymity, we cannot disclose the specific platform). To maintain a certain level of general capability for the LLM, we incorporated a substantial amount of open-source data, with domain-specific data accounting for 25% of the total. This domain-specific data mainly includes ASR and OCR from product live streams and short videos, user comments, and textual information about the products themselves. We set $p_{mar} = 0$, $p_{mrc} = 0.5$, $\alpha = 1$, where MAR has already degraded into AR. This is because: 1. we need to enhance the model's overall domain knowledge. 2. The downstream representation tasks only involve retrieval and classification, while MAR offers greater improvements for STS and clustering tasks.

The testing scenarios primarily focus on governance within the E-commerce Platform, with the ultimate goal of determining whether a given sample violates the platform's regulatory guidelines. The tasks can be categorized into three main types. The language model task primarily involves generative QA, where for each sample, the model is directly asked whether it violates a specific set of regulations. The retrieval task focuses on identifying samples that share the same type of violation as a given problematic sample. The classification task involves using a large language model (LLM) to offline extract text representations of a sample, which serve as input for a classification head to categorize the sample's violations.

[5]https://github.com/FlagOpen/FlagEmbedding
[6]https://huggingface.co/datasets/cfli/bge-full-data

G Full Result

Table 5 and table 6 shows the detailed scores of models trained with UniMAE and UniMAE + SimCSE on all of the 56 MTEB datasets.

Task	LLaMA3.2-1B	LLaMA3.2-3B	LLaMA3.1-8B
AmazonCounterfactualClassification.	76.58	80.66	75.01
AmazonPolarityClassification.	78.49	74.89	76.79
AmazonReviewsClassification.	40.26	39.98	40.34
ArguAna.	48.28	50.38	53.87
ArxivClusteringP2P.	45.54	46.99	47.45
ArxivClusteringS2S.	34.58	36.62	38.63
AskUbuntuDupQuestions.	55.73	57.02	55.00
BIOSSES.	79.97	84.28	85.12
Banking77Classification.	74.76	79.21	76.55
BiorxivClusteringP2P.	36.41	35.89	35.86
BiorxivClusteringS2S.	26.53	26.61	29.71
CQADupstackTexRetrieval.	18.02	18.95	17.59
ClimateFEVER.	21.99	19.03	22.36
DBPedia.	26.13	22.56	23.78
EmotionClassification.	44.82	48.18	42.86
FEVER.	40.32	35.14	37.44
FiQA2018.	24.50	26.62	25.70
HotpotQA.	43.58	49.07	50.59
ImdbClassification.	70.94	74.79	74.78
MSMARCO.	21.33	20.29	19.64
MTOPDomainClassification.	92.25	93.61	93.58
MTOPIntentClassification.	73.41	74.20	71.03
MassiveIntentClassification.	70.48	70.32	68.05
MassiveScenarioClassification.	74.79	75.03	74.79
MedrxivClusteringP2P.	29.50	26.77	29.28
MedrxivClusteringS2S.	22.85	23.61	25.98
MindSmallReranking.	29.92	29.14	32.91
NFCorpus.	26.40	28.92	28.06
NQ.	33.42	32.80	33.70
QuoraRetrieval.	80.71	81.28	82.67
RedditClustering.	43.36	43.28	48.13
RedditClusteringP2P.	58.50	59.13	60.81
SCIDOCS.	12.70	12.43	13.51
SICK-R.	71.41	67.37	68.52
STS12.	59.38	60.37	58.89
STS13.	75.44	75.00	73.64
STS14.	68.58	69.69	70.45
STS15.	78.73	79.10	78.93
STS16.	76.94	78.43	75.52
STS17.	83.87	80.87	83.13
STS22.	63.52	63.75	64.68
STSBenchmark.	73.52	71.77	75.59
SciDocsRR.	72.72	74.21	78.48
SciFact.	63.71	65.22	66.17
SprintDuplicateQuestions.	81.65	90.50	87.58
StackExchangeClustering.	57.92	63.54	63.80
StackExchangeClusteringP2P.	32.58	33.96	33.36
StackOverflowDupQuestions.	43.68	45.80	44.58
SummEval.	29.93	29.25	30.40
TRECCOVID.	44.99	40.94	44.71
Touche2020.	11.16	10.19	10.74
ToxicConversationsClassification.	69.44	72.38	69.21
TweetSentimentExtractionClassification.	60.38	58.43	58.90
TwentyNewsgroupsClustering.	34.79	32.05	40.71
TwitterSemEval2015.	61.70	60.61	62.45
TwitterURLCorpus.	84.37	83.82	84.65
AVG	52.81	53.30	53.87

Table 5: Unsupervised results of UniMAE transformed models on MTEB.

Task	LLaMA3.2-1B	LLaMA3.2-3B	LLaMA3.1-8B
AmazonCounterfactualClassification	72.39	78.01	67.97
AmazonPolarityClassification	76.94	79.38	77.04
AmazonReviewsClassification	37.80	40.27	40.27
ArguAna	41.26	49.76	52.12
ArxivClusteringP2P	47.43	48.03	49.04
ArxivClusteringS2S	39.29	40.66	45.22
AskUbuntuDupQuestions	57.32	59.21	59.41
BIOSSES	83.21	85.32	87.23
Banking77Classification	78.85	81.68	79.57
BiorxivClusteringP2P	37.60	35.43	36.64
BiorxivClusteringS2S	33.39	33.62	34.74
CQADupstackTexRetrieval	16.13	21.40	19.48
ClimateFEVER	15.81	19.54	22.67
DBPedia	24.52	24.85	29.53
EmotionClassification	48.30	50.47	45.34
FEVER	24.78	37.28	55.28
FiQA2018	23.43	27.64	28.81
HotpotQA	33.02	46.53	53.72
ImdbClassification	74.79	77.36	76.43
MSMARCO	16.98	19.99	21.56
MTOPDomainClassification	93.54	94.85	93.28
MTOPIntentClassification	70.46	75.96	71.61
MassiveIntentClassification	71.56	71.67	70.35
MassiveScenarioClassification	76.89	78.67	77.22
MedrxivClusteringP2P	30.06	29.83	30.58
MedrxivClusteringS2S	29.34	28.56	30.46
MindSmallReranking	32.83	31.90	33.36
NFCorpus	27.36	31.99	27.99
NQ	27.32	30.93	31.83
QuoraRetrieval	85.61	85.71	85.97
RedditClustering	54.29	57.34	57.39
RedditClusteringP2P	57.13	59.43	61.87
SCIDOCS	14.53	15.74	16.89
SICK-R	73.99	70.22	72.73
STS12	65.70	63.51	62.53
STS13	81.08	79.88	79.11
STS14	75.14	74.36	74.75
STS15	83.15	82.50	81.50
STS16	81.18	81.66	80.75
STS17	88.88	86.53	85.55
STS22	55.87	61.67	58.93
STSBenchmark	81.32	79.68	78.99
SciDocsRR	80.06	80.85	84.22
SciFact	61.10	65.41	67.24
SprintDuplicateQuestions	85.35	91.35	88.71
StackExchangeClustering	66.02	68.15	71.84
StackExchangeClusteringP2P	32.44	33.70	33.75
StackOverflowDupQuestions	46.71	49.69	50.67
SummEval	30.14	30.33	29.80
TRECCOVID	51.78	52.27	51.29
Touche2020	9.45	12.26	13.27
ToxicConversationsClassification	69.14	71.11	67.81
TweetSentimentExtractionClassification	60.45	59.74	58.39
TwentyNewsgroupsClustering	48.39	46.73	55.11
TwitterSemEval2015	65.35	65.59	67.52
TwitterURLCorpus	83.42	85.75	84.24
Average	54.11	56.11	56.60

Table 6: Unsupervised results of UniMAE+SimCSE transformed models on MTEB.