ULLME: A Unified Framework for Large Language Model Embeddings with Generation-Augmented Learning

Hieu Man[1], **Nghia Trung Ngo**[1], **Franck Dernoncourt**[2], **Thien Huu Nguyen**[1],
[1]Dept. of Computer Science, University of Oregon, OR, USA
[2]Adobe Research, USA
{hieum,nghian,thienn}@uoregon.edu, franck.dernoncourt@adobe.com

Abstract

Large Language Models (LLMs)[1] excel in various natural language processing tasks, but leveraging them for dense passage embedding remains challenging. This is due to their causal attention mechanism and the misalignment between their pre-training objectives and the text ranking tasks. Despite some recent efforts to address these issues, existing frameworks for LLM-based text embeddings have been limited by their support for only a limited range of LLM architectures and fine-tuning strategies, limiting their practical application and versatility. In this work, we introduce the Unified framework for Large Language Model Embedding (ULLME), a flexible, plug-and-play implementation that enables bidirectional attention across various LLMs and supports a range of fine-tuning strategies. We also propose Generation-augmented Representation Learning (GRL), a novel fine-tuning method to boost LLMs for text embedding tasks. GRL enforces consistency between representation-based and generation-based relevance scores, leveraging LLMs' powerful generative abilities for learning passage embeddings. To showcase our framework's flexibility and effectiveness, we release three pre-trained models from ULLME with different backbone architectures, ranging from 1.5B to 8B parameters, all of which demonstrate strong performance on the Massive Text Embedding Benchmark. Our framework is publicly available at: https://github.com/nlp-uoregon/ullme. A demo video for ULLME can also be found at https://rb.gy/ws1ile.

1 Introduction

For many years, the field of information retrieval has been dominated by a paradigm that relied heavily on pre-trained bidirectional encoders or encoder-decoders to obtain effective representation vectors for input texts (representation learning), e.g., BERT (Devlin et al., 2019) and T5 (Raffel et al., 2023). These architectures have played a pivotal role in advancing various language understanding tasks, including passage retrieval (Ni et al., 2022; Qu et al., 2021; Reimers and Gurevych, 2019), *inter alia*. However, recent research has witnessed a shift towards scaling representation learning methods to modern autoregressive language models (Muennighoff, 2022; Muennighoff et al., 2024; BehnamGhader et al., 2024). Leveraging the ongoing advancements in large language models (LLMs) with various sizes and domains, this approach has the potential to transform research in information retrieval, significantly improving performance on related tasks.

However, directly applying pre-trained LLMs to dense retrieval still presents numerous challenges. These challenges primarily stem from two factors: the inherent limitations of LLMs' causal attention mechanism which restricts the models' attention to only preceding tokens (Muennighoff, 2022; Springer et al., 2024), and the persistent misalignment between LLM pre-training objectives and text-ranking tasks (Ma et al., 2023; Muennighoff et al., 2024; BehnamGhader et al., 2024). To address these issues, researchers have developed methods to enable bidirectional attention within

Framework	#Supported LLMs	Supported Fine-tuning Strategy		
		SFT	DPO	Contrastive
SentenceTrasformers (Reimers and Gurevych, 2019)	>10	✗	✗	✗
SGPT (Muennighoff, 2022)	1	✗	✗	✓
RepLLaMA (Ma et al., 2023)	1	✗	✗	✓
Echo-Embedding (Springer et al., 2024)	2	✗	✗	✗
GritLM (Muennighoff et al., 2024)	2	✓	✗	✓
LLM2Vec (BehnamGhader et al., 2024)	3	✗	✗	✓
NV-Emb (Lee et al., 2024)	1	✗	✗	✓
ULLME (**our**)	>10	✓	✓	✓

Table 1: Comparisions between ULLME and other LLM-Embedding frameworks. For ULLME, the module combination enables many possible models and 10 is the number of models we have tested for usability.

[1]The definition of LLMs is vague. Here, we use "LLMs" to refer to models with more than 1 billion parameters. Moreover, in the scope of this work, we focus on decoder-only LLMs.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing
System Demonstrations, pages 230–239

LLMs by replacing the causal attention mask, which only allows attention to previous tokens, with an all-one mask that enables full contextual awareness. Furthermore, to better align the models with text retrieval tasks, researchers have employed fine-tuning strategies using retrieval-related data. However, as illustrated in Table1, existing frameworks for LLM-based representation learning have been limited in their scope, supporting only a narrow range of LLM architectures and fine-tuning strategies. This limitation highlights the need for a flexible and comprehensive framework that can accommodate diverse combinations of LLM backbones and fine-tuning approaches to facilitate full explorations of possibilities in different areas.

In this paper, we present ULLME, a versatile and extensible platform designed to advance the use of LLMs for dense retrieval. ULLME addresses the critical limitations of existing frameworks by offering a comprehensive, plug-and-play solution that seamlessly enables bidirectional attention across a array of diverse LLM families, including LLaMa, Mistral, Phi, Qwen, among others. Our framework's flexibility also extends beyond model compatibility, supporting a wide spectrum of fine-tuning strategies for LLM-based representation learning. As such, ULLME provides an unified framework for various LLM backbones and fine-tuning methodologies, allowing developers to comprehensively explore the full potential of LLMs in diverse embedding tasks, free from the constraints of implementation-specific restrictions.

In addition, existing frameworks for LLM-based text embeddings can be challenging for general users who are not familiar with training details like contrastive learning with large batch sizes and efficient fine-tuning. ULLME lowers these entry barriers by providing an efficient, user-friendly abstraction from those complexities, allowing users to focus on their data and tasks. For instance, ULLME's training processes are integrated with advanced techniques like GradCache (Gao et al., 2021a) and LoRa (Hu et al., 2022), enabling efficient contrastive learning and tuning with larger batch sizes, and sparing users from complicated configuration and testing. ULLME also comes with user-friendly features that make it easy to evaluate various fine-tuned LLMs using the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023), a comprehensive evaluation suite with numerous tasks for text embeddings.

Building upon the ULLME framework, we further introduce Generation-augmented Representation Learning (GRL), a novel fine-tuning strategy that leverages LLMs' generative capabilities for enhanced passage embedding. GRL bridges traditional dense retrieval methods with LLMs' inherent generation strengths through two key mechanisms: (i) Joint Training: we simultaneously fine-tune LLMs on passage generation and contrastive learning tasks; (ii) Generation-Guided Representation Learning: we propose to directly leverage the passage's generation probabilities of LLMs to enhance representation learning. This is achieved by encouraging consistency between the passage-query cosine similarities (derived from learned embeddings) and the passages' generation probability of LLMs given the queries. GRL thus effectively aligns the understanding of LLMs for text relevance with respect to both the embedding and generation spaces, leading to more nuanced and richer embeddings from LLMs.

To showcase the versatility and effectiveness of ULLME, we release three pre-trained LLM-Embedding models with different backbone architectures, ranging from 1.5B to 8B parameters, which deliver highly competitive results on MTEB. Our findings also highlight the advantages of our new fine-tuning method, GRL, which significantly outperforms the strong baselines, underscoring the potential of our framework to advance research and development in LLM-based embeddings.

2 Related Work

Our work is situated within the field of Information Retrieval (IR), specifically focusing on frameworks that leverage Large Language Models (LLMs) for Dense Retrieval.

LLMs for Dense Retrieval. Recent advancements in this area have primarily addressed two key challenges: (i): Overcoming LLMs' Causal Attention Limitations by developing methods to enable bidirectional attention within LLMs (Muennighoff, 2022; Muennighoff et al., 2024; BehnamGhader et al., 2024; Lee et al., 2024), allowing models to consider both past and future context when computing embeddings, and (ii): Aligning LLM Pretraining with Text Ranking by fine-tuning LLMs via contrastive learning (Ma et al., 2023; Wang et al., 2024; Lee et al., 2024). This process can also be augmented with additional objectives such as supervised fine-tuning (SFT) (Muennighoff et al., 2024) or mask-filling tasks (BehnamGhader

et al., 2024). An alternative approach proposed by Springer et al. (2024) involves a prompting method where the input sequence is duplicated, enabling each token to attend to future tokens and mitigating the contextualization issues inherent in causal attention. While these methods have shown promise, they generally do not explicitly enforce consistency between the model's understanding of relevance in both the embedding and generation spaces. This limitation restricts their ability to fully leverage the remarkable generative capabilities of LLMs for dense retrieval tasks. Our work, GRL, builds upon these foundations while addressing their limitations, introducing novel techniques to harmonize embedding-based and generation-based relevance scoring within a unified framework.

Frameworks of LLMs for Dense Retrieval. Existing frameworks for LLMs in Dense Retrieval have been constrained by their limited support for LLM architectures and fine-tuning strategies. As shown in Table1, SentenceTransformers(Reimers and Gurevych, 2019) supports various types of LLMs but is primarily designed for inference without allowing fine-tuning, limiting its applicability in advancing state-of-the-art dense retrieval methods. Some recent works (Muennighoff, 2022; Ma et al., 2023; Lee et al., 2024), such as **Echo** (Wang et al., 2024), **GritLM** (Muennighoff et al., 2024), **LLM2Vec** (BehnamGhader et al., 2024), and the models in the Hugging Face's MTEB leaderboard[2], have introduced implementations for LLM-based text embeddings. However, these approaches are often tailored to specific model architectures and training methods with hard-coded implementations, thus restricting their adaptability and use across different LLM architectures and fine-tuning strategies to meet diverse development and application demands. In contrast, our framework ULLME addresses these limitations by offering a flexible and extensible platform. ULLME can accommodate a diverse range of LLM backbones and supports various training approaches, making it highly versatile and broadly applicable.

3 ULLME - Unified framework for Large Language Model Embedding

We present an overview of our ULLME framework in Section 3.1 while Section 3.2 details the key technical methods.

[2] https://huggingface.co/spaces/mteb/leaderboard

```python
from ullme.models import ULLME

model = ULLME(
    model_name_or_path="mistralai/Mistral-7B-v0.1",
    model_backbone_type="mistral",
    lora_name="ullme-mistral",
    loar_r=16,
    lora_alpha=32,
    )
input_sentence = "This a example sentence."
model_inputs = model.tokenizer(
    [input_sentence],
    return_tensors='pt'
    )
model_output = model(
    input_ids=model_inputs['input_ids'],
    attention_mask=model_inputs['attention_mask'],
    is_generate=False
    )
>> {'rep': (1, hidden_dim)}
```

Listing 1: Extending bidirectional attention for LLMs via ULLME.

3.1 Overview

ULLME addresses the limitations of existing LLM-based dense retrieval frameworks by offering a flexible and comprehensive solution. The framework operates in three main stages. First, it enables bidirectional attention within LLMs by replacing the causal attention mask with a bidirectional one. This crucial modification extends the models' ability to consider both past and future context when generating embeddings, significantly enhancing its capacity for dense retrieval tasks. The transformed model is then returned as a PyTorch object, providing users with the flexibility to integrate it into various frameworks or pipelines. We will elaborate on this process in Section 3.2.1. Second, ULLME supports a diverse array of fine-tuning strategies, including Contrastive Learning, Supervised Fine-tuning (SFT), Direct Preference Optimization (DPO), and our novel Generation-augmented Representation Learning (GRL). This versatility allows for tailored optimization across a wide spectrum of retrieval tasks and domains, as detailed in Section 3.2.2. Finally, the framework streamlines the evaluation process by incorporating direct support for model validation using the Massive Text Embedding Benchmark (MTEB) library (Section 3.3). This integration facilitates comprehensive assessment across numerous retrieval and embedding tasks. By seamlessly combining these elements, ULLME provides an extensive toolkit for leveraging LLMs in diverse dense retrieval tasks,

encompassing everything from initial model adaptation to fine-tuning and evaluation. Our comprehensive approach aims to accelerate research and development for of LLM-based dense retrieval, offering researchers and practitioners a comprehensive platform for innovation and advancement.

3.2 Key Features

3.2.1 Enabling Bidirectional Attention

To enable bidirectional attention in LLMs, ULLME requires only minimal code modifications, as illustrated in Listing 1. The framework's user-friendly design allows for easy initialization with various LLM backbones by simply specifying the "`model_name_or_path`" and "`model_backbone_type`" parameters. ULLME seamlessly integrates with Hugging Face Transformers, loading pre-trained LLMs directly from their repository. Additionally, our framework supports parameter-efficient fine-tuning through Low-Rank Adaptation (LoRA) (Hu et al., 2022), offering flexibility in model adaptation. Once initialized, the model can be used to compute sequence representations. The "`is_generate`" parameter plays a crucial role in controlling the attention mechanism: when set to "`False`", the model employs bidirectional attention, optimizing it for dense retrieval tasks, while "`True`" reverts the model to causal attention, mimicking the standard Hugging Face Transformer model output. This dual functionality allows ULLME to serve both as an advanced specialized embedding model and as a language model when needed, providing developers with a flexible tool that can conveniently transition between bidirectional and causal attention modes. ULLME provides various methods for extracting text embeddings from LLMs, such as using representations from the first token, last token, mean, or weighted mean pooling. However, it defaults to averaging the representation vectors from the final layers (mean) for better performance on our datasets.

3.2.2 Fine-tuning Strategies

Our ULLME framework supports multiple fine-tuning strategies, as illustrated in Listing 2.

Contrastive Learning. ULLME's Contrastive Learning objective utilizes in-batch negatives (Chen et al., 2020; Gao et al., 2021b). The contrastive loss is formally defined as: $\mathcal{L}_{CL} = -\log \frac{\exp\left(s_{rt}(q,p^+)\right)}{\exp\left(s_{rt}(q,p^+)\right)+\sum_{p^-\in B}\exp\left(s_{rt}(q,p^-)\right)}$.

Here, B represents a mini-batch, q is the input query, p^+ denotes the positive (relevant) passage, and p^- represents negative (non-relevant) passages sampled from the current training mini-batch. The function $s_{rt}(q,p)$ computes the relevance score between a query and a passage using cosine similarity of the induced representations for q and p. To enhance the effectiveness of Contrastive Learning, especially under limited GPU memory constraints, ULLME incorporates advanced techniques such as GradCache (Gao et al., 2021a) and cross-device contrastive loss computation. These optimizations allow for efficient training with larger batch sizes and more diverse negative samples, which are crucial for learning high-quality representations.

```python
from ullme.trainer import GradCacheTrainer

trainer = GradCacheTrainer(
    con_loss_type='NTXentLoss',
    gen_loss_type='dpo', # 'sft'
    use_kl_loss=True
)
trainer.fit_epoch(
    model=model,
    train_loader=train_dataloader,
)
```

Listing 2: Finetuning LLMs for text embedings via ULLME.

Supervised Fine-tuning (SFT). In addition to contrastive learning, ULLME supports SFT, a strategy that enhances LLMs' ability to generate high-quality passages in response to queries. ULLME implements SFT using a next-word prediction objective: $\mathcal{L}_{SFT} = -\frac{1}{N}\sum_{i=1}^{N}\log \pi_\theta(w_i|w_{<i},q)$. Here, N is the length of the positive passage p^+, w_i is the i-th token in p^+, and $\pi_\theta(w|x)$ is the conditional likelihood of w given x, computed by the LLM θ. Importantly, during SFT loss computation, ULLME reverts to using causal attention, mirroring standard LLM behavior.

Direct Preference Optimization (DPO). ULLME incorporates Direct Preference Optimization (DPO) (Rafailov et al., 2023) as an advanced fine-tuning strategy, offering an alternative to traditional Supervised Fine-tuning (SFT). DPO has demonstrated superior effectiveness in LLM fine-tuning. Moreover, the DPO approach inherently accounts for both preferred and rejected outputs, making it intuitively more suitable for aligning models with text-ranking objectives compared to SFT. In ULLME's implementation, the ground-truth relevant passage p^+ for a query q is treated as

the preferred output, while negative and irrelevant passages p^- are considered dispreferred. The DPO loss function is designed to encourage the model to assign higher generation probabilities to p^+ compared to any p^-: $\mathcal{L}_{DPO} = -\log\sigma\left(\beta\log\frac{\pi_\theta(p^+|q)}{\pi_{ref}(p^+|q)} - \beta\log\frac{\pi_\theta(p^-|q)}{\pi_{ref}(p^-|q)}\right)$. In this formulation, σ represents the sigmoid function, β is a scaling factor, and $\pi_{ref}(p|q)$ denotes the conditional likelihood computed by the original pre-trained LLM (the reference model).

In addition to the standard DPO formulation, ULLME includes implementations of advanced variants such as Kahneman-Tversky Optimization (KTO) (Ethayarajh et al., 2024) and Contrastive Preference Optimization (CPO) (Xu et al., 2024). The modular architecture of ULLME facilitates the seamless integration of new preference optimization techniques as they emerge, ensuring that the framework remains at the forefront of LLM fine-tuning advancements. Finally, to maintain consistency with the model's pre-training paradigm, ULLME employs causal attention when computing the DPO loss, similar to the approach used in SFT.

Generation-augmented Representation Learning (GRL). ULLME further introduces a novel fine-tuning strategy GRL that explicitly aligns the LLMs' understanding of passage-query text relevance in embedding and generation spaces to boost representation learning. As such, GRL first computes a generation-based relevance score $s_{gen}(q,p)$ utilizing the conditional generation likelihood of a passage candidate p given input query q from LLMs: $s_{gen}(q,p) = \frac{1}{t}\sum_{i=1}^{t}\log\pi_\theta(w_i|w_{<i},q)$, where t is the length of p and w_i is the i-th token in p.

Next, we seek to recognize the consistency of the query-passage relevance scores obtained from the representations (i.e., $s_{rt}(q,p)$) and the generation likelihood (i.e., $s_{gen}(q,p)$). Particularly, let U be the set of m candidate passages for q. For each candidate passage $p_i \in U$, we compute $s_{rt}(q,p_i)$ and $s_{gen}(q,p_i)$, then normalize these scores to obtain the representation and generation relevance distributions over U: $P_{rt}(q,p_i) = \frac{\exp(s_{rt}(q,p_i))}{\sum_{p'\in U}\exp(s_{rt}(q,p'))}$ and $P_{gen}(q,p_i) = \frac{\exp(s_{gen}(q,p_i))}{\sum_{p'\in U}\exp(s_{gen}(q,p'))}$.

Afterward, we minimize the KL divergence between their distributions: $\mathcal{L}_{KL} = \sum_{p\in U} P_{rt}(q,p)\log\frac{P_{rt}(q,p)}{P_{gen}(q,p)}$, serving as a training signal to enrich representation learning for LLMs.

```python
from ullme.models import WrappedULLME
from ullme.eval import eval_mteb_dataset

model = WrappedULLME(
    model_name_or_path="mistralai/Mistral-7B-v0.1",
    model_backbone_type="mistral",
    lora_name="ullme-mistral",
    loar_r=16,
    lora_alpha=32,
    model_checkpoint="path/to/your/checkpoint"
    )
eval_result = eval_mteb_dataset(
    model=model,
    dataset_name='MSMARCO',
    langs=['eng'],
    )
>> {'eng': 35.8}
```

Listing 3: Evaluation on MTEB dataset via ULLME.

Finally, the overall training loss for GRL combines the contrastive loss $\mathcal{L}_{CL}$, the direct preference optimization loss $\mathcal{L}_{DPO}$, and the KL-divergence loss $\mathcal{L}_{KL}$: $\mathcal{L}_{GRL} = \lambda_{CL}\mathcal{L}_{CL} + \lambda_{DPO}\mathcal{L}_{DPO} + \lambda_{KL}\mathcal{L}_{KL}$, where λ_{CL}, λ_{DPO}, and λ_{KL} are weighting hyperparameters.

3.3 Evaluation Process

ULLME streamlines the evaluation process by integrating direct support for evaluating LLM-based text embedding models over MTEB[3], a widely-used Massive Text Embedding Benchmark with diverse tasks and datasets. This integration facilitates comprehensive model development with different methods and extensive assessment across numerous retrieval and embedding tasks in a single framework. ULLME wraps a fine-tuned model into a "`WrappedULLME`" instance, ensuring compatibility with MTEB's requirements for direct evaluation. In addition to supporting ULLME's fine-tuned models, our evaluation function is designed to perform seamlessly with most LLM models available in the Hugging Face ecosystem, including the latest LLM-Embedding models in the MTEB leaderboard. Users can easily specify the desired model through the "`model_name_or_path`" parameter, enabling effortless evaluation of various LLMs without the need for extensive configuration. ULLME allows users to select specific datasets and language subsets for evaluation. The evaluation results are reported using MTEB's predefined main scores of the corresponding dataset, ensuring standardized and comparable metrics across different models, as demonstrated in Listing 3.

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

4 Experiments

Our ULLME framework supports various LLM architectures and fine-tuning strategies for text embeddings with convenient interface. To highlight the framework's flexibility, we demonstrate the operations of ULLME with three different base LLMs ranging from 1.5B to 8B parameters: Phi-1.5B (Li et al., 2023), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), and Meta-LLama3-8B-Instruct (AI@Meta, 2024). For each LLM, we evaluate ULLME's performance for different combinations of attention and fine-tuning approaches, including: **Base**: Original causal model, **Causal + CL**: Causal model fine-tuned with Contrastive Learning, **Bi + CL**: Bidirectional-enabled model fine-tuned with Contrastive Learning, and **Bi + CL + SFT**: Bidirectional-enabled model fine-tuned with Contrastive Learning and SFT. In addition, we report the performance of our Generation-augmented Representation Learning (GRL) method for fine-tuning LLMs in ULLME, featuring the full model GRL and **GRL**$_{SFT}$, a variant of GRL that replaces DPO with SFT for tuning. Finally, we compare the performance of ULLME's models with recent state-of-the-art methods for LLM-based text embeddings, including **Echo** (Wang et al., 2024) and **LLM2Vec** (BehnamGhader et al., 2024).

Settings. Following prior work (Qu et al., 2021; Ren et al., 2021; Ma et al., 2023), we use a curated subset of the MSMARCO dataset (Bajaj et al., 2018) for model training. MTEB datasets are employed for evaluation. To train the models, we utilize LoRA (Hu et al., 2022) with $r = 16$ and $\alpha = 32$, and enable various optimization techniques, i.e., GradCache, gradient checkpointing, mixed precision training, and FSDP (Zhao et al., 2023), to minimize GPU memory requirements. We utilize the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 2e-4 and a batch size of 512 with the number of hard negative passages per example was set to 8. We train the models for one epoch on MSMARCO. The weights for the GRL loss components include $\lambda_{CL} = \lambda_{KL} = 1$ and $\lambda_{DPO} = 0.5$. The scaling factor β in the DPO loss was set to 0.1.

Results. Table 2 showcases the performance of various models on the MTEB datasets. Compared to previous methods Echo and LLM2Vec, it is clear that our ULLME framework can be used to train diverse and competitive LLM-based embedding models for different base LLMs and tasks in MTEB.

	Phi 1.5	Mistral-2-7B	LlaMa-3-8B
Echo*	36.00	50.26	51.11
LLM2Vec*	54.47	57.47	58.04
Base	31.15	42.31	42.33
Causal + CL	51.83	54.03	54.68
Bi + CL	52.70	55.41	55.86
Bi + CL + SFT	53.88	57.01	56.83
GRL$_{SFT}$	55.01	58.37	57.50
GRL (ours)	**55.76**	**59.50**	**59.27**

Table 2: Model performances on MTEB datasets using MSMARCO for training data. The numbers are averaged over 56 datasets of MTEB, covering diverse tasks such as Retrieval, Reranking, Clustering, Pair Classification, Classification, Semantic Textual Similarity, and Summarization. The best results are in bold and * indicates our implementation/reproduced results using the same training data. Detailed performance for all datasets in MTEB is reported in Table 3.

Among various architectures in ULLME, we observe that the combination of contrastive learning and SFT leads to better performance than the individual techniques, demonstrating their complementary benefits for LLM-based embeddings. Notably, our proposed Generation-augmented Representation Learning (GRL) method in ULLME consistently outperforms the best baseline, LLM2Vec, across different base models ranging from 1.5B to 8B parameters. This highlights the effectiveness of using generation probabilities to guide representation learning in GRL. Finally, we note that the inference time of the fine-tuned models with ULLME is comparable to the original LLMs, processing 16K, 12K, and 12.8K tokens per second for Phi-1.5B, Mistral-7B-Instruct-v0.2, and Meta-LLama3-8B-Instruct, respectively.

5 Conclusion

We introduce ULLME (Unified framework for Large Language Model Embedding), a comprehensive and flexible toolkit for leveraging LLMs for text embeddings and dense retrieval tasks. Our work addresses critical limitations in existing frameworks for LLM embeddings by providing support for various LLM architectures, fine-tuning strategies, and benchmark evaluation within a single, user-friendly framework. Our experimental results demonstrate the effectiveness of ULLME, particularly the GRL strategy, in improving dense retrieval performance across various LLM scales and tasks. Our potential future directions include exploration of better techniques to leverage the generative and discriminative capabilities of LLMs,

and extension of the framework to support emerging LLM architectures and training paradigms. We anticipate that ULLME will facilitate broader applications of LLM embeddings in downstream tasks, ranging from deep context understanding requirements like sentiment analysis (Gupta et al., 2024) to text style comprehension tasks such as authorship attribution (Rivera-Soto et al., 2021; Man and Huu Nguyen, 2024), thereby contributing to the advancement of natural language processing and information retrieval fields.

Acknowledgements

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112, the NSF grant CNS-1747798 to the IUCRC Center for Big Learning, and the NSF grant # 2239570. This research is also supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract 2022-22072200003. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government.

References

AI@Meta. 2024. Llama 3 model card.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset. *Preprint*, arXiv:1611.09268.

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *Preprint*, arXiv:2402.01306.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021a. Scaling deep contrastive learning batch size under memory limited setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 316–321, Online. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. 2024. Comprehensive study on sentiment analysis: From rule-based to modern llm based system. *Preprint*, arXiv:2409.09989.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *Preprint*, arXiv:2405.17428.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: **phi-1.5** technical report. *arXiv preprint arXiv:2309.05463*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning llama for multi-stage text retrieval. *Preprint*, arXiv:2310.08319.

Hieu Man and Thien Huu Nguyen. 2024. Counterfactual augmentation for robust authorship representation learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2347–2351, New York, NY, USA. Association for Computing Machinery.

Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *Preprint*, arXiv:2202.08904.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. *Preprint*, arXiv:2402.09906.

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

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland. Association for Computational Linguistics.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rafael A. Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y. Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. 2021. Learning universal authorship representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 913–919, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

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

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *Preprint*, arXiv:2401.08417.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. 2023. Pytorch fsdp: Experiences on scaling fully sharded data parallel. *Preprint*, arXiv:2304.11277.

A Detailed Performance on MTEB

We present the full performance of the three ULLME-released models – Phi-1.5 (Li et al., 2023), Mistral-2-7B-instruct (Jiang et al., 2023), and LLaMa-3-B-instruct (AI@Meta, 2024) – across the MTEB datasets in Table 3.

Task	Phi 1.5	Mistral-2-7B	LlaMa-3-8B
AmazonCounterfactualClassification	67.79	75.28	73.69
AmazonPolarityClassification	72.03	77.40	78.51
AmazonReviewsClassification	35.58	39.78	38.31
Banking77Classification	84.24	84.57	84.76
EmotionClassification	45.83	45.02	49.48
ImdbClassification	66.73	72.47	74.97
MassiveIntentClassification	70.43	73.41	73.1
MassiveScenarioClassification	76.75	78.28	78.59
MTOPDomainClassification	92.58	94.72	94.70
MTOPIntentClassification	69.63	77.05	73.49
ToxicConversationsClassification	66.26	60.62	64.21
TweetSentimentExtractionClassification	55.92	55.99	56.63
ArxivClusteringP2P	42.29	46.97	46.46
ArxivClusteringS2S	31.65	39.92	37.91
BiorxivClusteringP2P	36.25	38.18	38.35
BiorxivClusteringS2S	30.46	31.48	30.32
MedrxivClusteringP2P	31.82	32.32	32.19
MedrxivClusteringS2S	30.18	26.95	26.01
RedditClustering	49.31	41.45	41.96
RedditClusteringP2P	55.85	62.26	61.64
StackExchangeClustering	60.6	62.44	61.06
StackExchangeClusteringP2P	31.79	32.99	33.77
TwentyNewsgroupsClustering	42.95	38.52	41.32
SprintDuplicateQuestions	92.78	92.2	94.73
TwitterSemEval2015	59.19	67.35	69.0
TwitterURLCorpus	85.06	86.81	85.61
AskUbuntuDupQuestions	59.23	63.62	63.43
MindSmallReranking	31.70	32.30	31.66
SciDocsRR	79.29	83.47	81.42
StackOverflowDupQuestions	48.61	52.56	52.38
ArguAna	55.06	45.93	46.78
ClimateFEVER	22.28	28.10	22.22
CQADupstackTexRetrieval	22.39	25.84	28.30
DBPedia	30.45	46.55	46.36
FEVER	58.11	79.39	61.52
FiQA2018	32.25	42.97	42.28
HotpotQA	48.44	64.04	67.41
MSMARCO	28.65	34.22	35.65
NFCorpus	34.54	39.37	39.37
NQ	38.37	60.73	61.36
QuoraRetrieval	86.49	88.33	87.75
SCIDOCS	16.46	21.00	21.13
SciFact	63.41	72.86	72.38
Touche2020	16.56	30.52	27.13
TRECCOVID	54.21	84.74	83.56
BIOSSES	85.35	78.64	83.74
SICK-R	70.49	70.31	69.11
STS12	71.83	67.25	69.95
STS13	80.05	82.35	79.58
STS14	74.19	75.04	73.67
STS15	83.0	82.69	83.47
STS16	79.77	81.15	81.58
STS17	88.49	86.38	86.3
STS22	67.77	68.54	67.35
STSBenchmark	80.81	78.21	80.25
SummEval	30.61	30.56	31.10
Average	55.76	59.50	59.27

Table 3: Performance of ULLME's released models on full MTEB benchmark using MSMARCO as training data.