Enhancing Lexicon-Based Text Embeddings with Large Language Models

Yibin Lei[1], Tao Shen[2], Yu Cao[3], Andrew Yates[1]
[1]University of Amsterdam [2]University of Technology Sydney [3]Tencent IEG
{y.lei, a.c.yates}@uva.nl, tao.shen@uts.edu.au,
rainyucao@tencent.com

Abstract

Recent large language models (LLMs) have demonstrated exceptional performance on general-purpose text embedding tasks. While dense embeddings have dominated related research, we introduce the first **L**exicon-based **E**mbeddi**N**g**S** (LENS) leveraging LLMs that achieve competitive performance on these tasks. Regarding the inherent tokenization redundancy issue and unidirectional attention limitations in traditional causal LLMs, LENS consolidates the vocabulary space through token embedding clustering, and investigates bidirectional attention and various pooling strategies. Specifically, LENS simplifies lexicon matching by assigning each dimension to a specific token cluster, where semantically similar tokens are grouped together, and unlocking the full potential of LLMs through bidirectional attention. Extensive experiments demonstrate that LENS outperforms dense embeddings on the Massive Text Embedding Benchmark (MTEB), delivering compact feature representations that match the sizes of dense counterparts. Notably, combining LENS with dense embeddings achieves state-of-the-art performance on the retrieval subset of MTEB (i.e. BEIR).

1 Introduction

Text embeddings are vector representations of text that power a wide range of applications, including retrieval, question answering, semantic textual similarity, and clustering. Recent advances in LLMs have shown that a single model can generate embeddings excelling across diverse tasks, highlighting their versatility (Li et al., 2024; BehnamGhader et al., 2024; Wang et al., 2023; Muennighoff et al., 2024; Meng et al., 2024; Lee et al., 2024a).

While dense embeddings that encode texts into low-dimensional, real-valued latent semantic spaces dominate recent research, lexicon-based embeddings (Formal et al., 2021b,a; Shen et al., 2023a; Lassance et al., 2024) offer distinctive advantages. These high-dimensional representations, where each dimension corresponds to a specific token of the vocabulary, align more closely with the pre-training objectives of language models due to their shared use of the vocabulary space and the language modeling head (Shen et al., 2023a). Recent studies have demonstrated that lexicon-based embeddings can surpass their dense counterparts, utilizing masked language models under specific control (Déjean et al., 2023). Additionally, lexicon-based embeddings can offer better transparency, providing clearer insights into the model's decisions via the weight of each token. Moreover, the combination of dense and lexicon-based embeddings has also been proven to be promising in prior studies, as they effectively complement each other (Lin, 2021; Shen et al., 2023b).

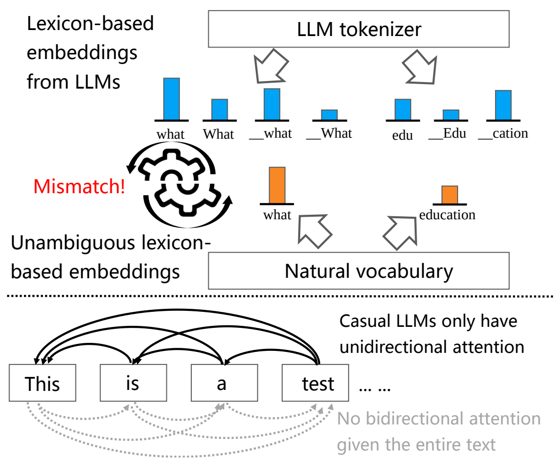

Figure 1: The redundancy and noise in LLM tokenizers, as well as the absence of bidirectional dependencies in causal LLMs motivate LENS.

Despite these benefits, lexicon-based embeddings remain underexplored beyond retrieval tasks. To unlock their full potential in more scenarios, it is essential to address the challenges posed by LLMs, as shown in Fig. 1. The first one is the inherent redundancy of LLM vocabularies. Since most modern tokenizers rely on subword tokenization (e.g., "education" is split into "edu" and "cation"), it fragments the entire vocabulary space (Soler et al., 2024). And semantically equivalent tokens can appear in multiple forms in the tokenizer (e.g., "what", "What", " what" and "review", "reviews"), introducing inconsistencies and difficulties in lexicon matching. Consequently, recent studies indicate that replacing the original tokenization of BM25 (Robertson et al., 1995) with the XLM-R tokenizer (Conneau et al., 2020) can lead to a significant performance drop due to the noisier vocabulary (Chen et al., 2024). The second challenge is that LLMs typically employ unidirectional attention during pre-training, where tokens can only attend to preceding tokens. This limitation prevents each token from fully leveraging the surrounding context, which is crucial as lexicon-based embeddings are always derived from the outputs of all tokens.

To address these challenges, we first explore the potential of LLMs generating embeddings where each dimension corresponds to a token cluster instead of the traditional single token, with each cluster grouping tokens that share similar meanings or stem from the same lexeme. To achieve this, we utilize a simple yet effective approach that directly clusters the token embeddings and leverages the centroids of these clusters as the new token embeddings for the language modeling head. As shown in Table 4, the resulting clusters naturally group tokens with similar meanings, forming more coherent and compact embeddings. At the meanwhile, these cluster-based embeddings can achieve the equivalent feature size as dense embeddings (e.g., 4,000d), which is much smaller than previous lexicon-based embeddings. Such a property not only i) facilitates the integration of LENS into existing dense frameworks like FAISS, freeing us from the sparsity constraints that, while essential for efficient retrieval, can limit expressiveness and effectiveness of models (Formal et al., 2024), but also ii) eliminates computational overhead in tasks such as clustering and classification, where inverted indices cannot be used.

Furthermore, to address the interior LLM architecture drawbacks, we also conduct extensive investigations into modifying the model frameworks. Given the recent studies highlight the significant impact of attention mechanisms and pooling strategies on dense embeddings (Li et al., 2024; Muennighoff et al., 2024; BehnamGhader et al., 2024; Lee et al., 2024a), we incorporate variants of these two factors in our framework to examine how they affect lexicon-based embeddings. Contrary to prior findings (Li et al., 2024), which suggest that preserving the original architecture of LLMs typically yields optimal performance for dense embeddings, our results indicate that bidirectional attention is critical for achieving superior performance with lexicon-based embeddings.

Built on these techniques, we introduce LENS, a framework designed to generate low-dimensional lexicon-based embeddings that achieve impressive results across a variety of tasks. Specifically, our experiments demonstrate that LENS outperforms dense embeddings on the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023), achieving state-of-the-art (SOTA) zero-shot performance among models trained exclusively on public data, as of December 1, 2024. Qualitative examples also illustrate that LENS produces grounded and meaningful representations. Further analysis demonstrates that LENS, even when using 2000 clusters, still outperforms embeddings that leverage the original vocabulary space. Moreover, combining LENS with dense embeddings achieves SOTA performance on the retrieval subset of MTEB (specifically, BEIR).

2 Related Work

Lexicon-Based Embeddings. Lexicon-based embeddings assign each dimension of the embedding vector to a specific token in the vocabulary. With the advancements in masked language models, recent studies have demonstrated that lexicon-based embeddings (Mallia et al., 2021; Lin and Ma, 2021; Zhuang and Zuccon, 2021; Formal et al., 2021b,a; Shen et al., 2023a; Nguyen et al., 2023; Lassance et al., 2024) can deliver superior performance. Among these approaches, SPLADE (Formal et al., 2021b,a; Lassance et al., 2024) stands out as one of the most effective methods, often outperforming dense embeddings (Déjean et al., 2023). Moreover, lexicon-based embeddings have been shown to complement dense embeddings, with their combination yielding substantial performance improvements (Chen et al., 2024; Shen et al., 2023b; Lin, 2021). Despite these advances, research on lexicon-based embeddings has largely focused on retrieval tasks, leaving other applications such as clustering and classification relatively underexplored.

LLM-Based Embeedings. As decoder-only LLMs continue to advance, recent work has investigated their potential for generating dense text embeddings capable of performing well across different tasks. To align LLMs with text embedding tasks, LLM2Vec (BehnamGhader et al., 2024) employs masked next-token prediction training and unsupervised contrastive learning, while LLaRA (Li et al., 2023a) leverages an auto-encoding objective to enhance embedding quality. Recent efforts, such as E5-Mistral (Wang et al., 2023) and Gecko (Lee et al., 2024b), focus on improving embedding models by using LLMs to generate diverse training data. Additionally, GRIT (Muennighoff et al., 2024) explores the combination of contrastive learning and language modeling objectives to train a single LLM that performs well on both embedding and generation tasks. Meanwhile, studies (Muennighoff et al., 2024; Lee et al., 2024a; BehnamGhader et al., 2024; Li et al., 2024) highlight the significant influence of architectural choices on embedding model performance, with findings (Li et al., 2024) indicating that retaining the original unidirectional attention often yields the best results.

Research on leveraging LLMs for lexicon-based embeddings remains limited. PromptReps (Zhuang et al., 2024) and Mistral-SPLADE (Doshi et al., 2024) use prompt engineering to generate lexicon-based embeddings from LLMs. However, these methods often perform worse than their dense counterparts, introduce additional computational overhead, and are limited to exploring only retrieval tasks.

3 Methodology

In this section, we first introduce preliminaries for a better understanding of the design of our framework, then formally describe the details of LENS.

3.1 Preliminaries

3.1.1 Lexicon-Based Embeddings Using Masked Language Models

SPLADE (Formal et al., 2021b,a; Lassance et al., 2024) is a representative method that utilizes Masked Language Models (MLMs) and regards the logits from the masked language modeling head as lexicon-based embeddings, leveraging the bidirectional attention. The MLM produces a sequence of logits $L = (l_1, l_2, \ldots, l_n), l_i \in \mathbb{R}^{|V|}$ given the input sequence, where $|V|$ is the vocabulary size. Each logit value l_{ij} represents the likelihood of the vocabulary token j being relevant to the position i. Specifically, these scores are produced by the language modeling head, which maps the output hidden states to the vocabulary space using the token embedding matrix.

To obtain the lexicon-based embeddings, SPLADE first applies a log-saturation transformation to the logits to scale the weight and enforce it as non-negative,

$$w_{ij} = \log\left(1 + ReLU(l_{ij})\right). \tag{1}$$

Then it performs max-pooling across logits of all tokens to derive the final weight for each vocabulary token,

$$w_j = \max_{i \in n} w_{ij}. \tag{2}$$

Despite its proven effectiveness, former research on lexicon-based embeddings using MLMs primarily focused on small-scaled models, leaving the performance of larger models mostly unexplored.

3.1.2 Lexicon-Based Embeddings Using Causal Language Models

Motivated by the growing capability of larger-scaled models, recent works have begun to use causal language models with significantly more parameters, such as LLaMA (Touvron et al., 2023) and Mistral (Jiang et al., 2023), to derive lexicon-based embeddings. Two notable methods are **PromptReps** (Zhuang et al., 2024) and **Mistral-SPLADE** (Doshi et al., 2024), which employ prompts to alleviate the limitations brought by the unidirectional attention.

PromptReps enables LLMs to generate both dense and lexicon-based embeddings through carefully designed prompts such as *"This sentence [INPUT] means in one word:"*. Dense embeddings are derived from the hidden states of the final token ", and lexicon-based ones are the logits for the next token prediction. Nevertheless, such a method relying solely on prompt causes a substantial performance drop of lexicon-based embeddings compared to their dense counterparts, e.g., MRR@10 of 34.15 vs. 41.86 on the MS MARCO dataset.

Mistral-SPLADE adapts SPLADE to large causal models like Mistral by using echo prompting (Springer et al., 2024). It enables full-context visibility of each token by duplicating the input sequence and regards the representations of the second occurrence as the output. Despite getting advancements on BEIR benchmark, it still lags behind dense embeddings like E5-Mistral (Wang et al., 2023) and LLM2Vec (BehnamGhader et al., 2024), demonstrating that lexicon-based embeddings using large models cannot solely rely on prompting.

Hence, LENS systematically investigate the architecture of LLMs, including attention mechanisms and pooling methods, rather than exterior prompting. We try to unlock the full potential of LLMs for lexicon-based embeddings, not only on retrieval tasks that have been widely examined before, but also on clustering and classification tasks which remain unexplored.

3.1.3 Tokenization in LLMs

LLM tokenizers, though designed to cover all possible text forms for the language modeling objective, may hinder the effectiveness of lexicon-based embedding. i) Extra redundancy can be introduced under the same lexeme and further affect the token matching. E.g., "What", "what", and " what" can be regarded as distinct tokens due to differences in case or whitespace, even though they represent the same word. ii) Subword fragmentation (Soler et al., 2024) split a common word into pieces like "education" into "edu" and "cation", posing additional matching complexity. iii) Tokenizers trained on large corpora often include rare tokens, which inflate the vocabulary size and make the embedding larger and slower to match.

Therefore, instead of directly using the original language modeling head, we simply cluster original tokens to form clusters and use their centroid embeddings to replace the original token embeddings of the language modeling head. This approach reduces the redundancy by merging related tokens and decreases the size of embeddings by using a smaller clustered vocabulary.

3.2 Framework of LENS

After discussing the background, we introduce the framework of our method, as shown in Fig. 2.

3.2.1 Architecture Design

Language Modeling Head. Motivated by the redundancy and noise in LLM tokenizer mentioned above, LENS assigns weights to groups of tokens with similar meanings, whose effectiveness has been verified in Zhang et al. (2024). Specifically, we apply KMeans clustering (Hartigan and Wong, 1979) to the token embeddings from the language modeling head, where k is our desired lexicon-

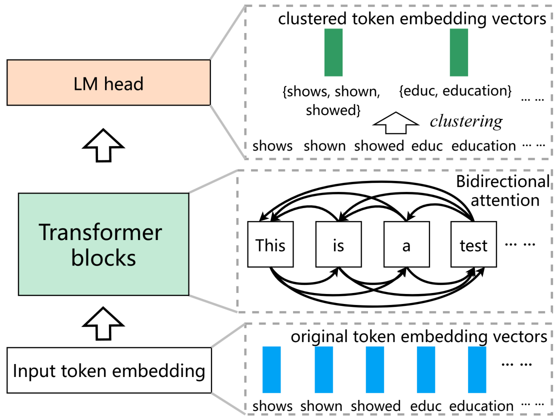

Figure 2: The model framework of LENS.

based embedding size. Then the original token embeddings in the LM head are replaced by the cluster centroids, while the input token embeddings remain unchanged. Such a substitution reduces the dimensionality of the lexicon-based embeddings, as the logits now represent scores over fewer clusters rather than the original huge vocabulary. Check Table 4 and Appendix A.1 for detailed cluster results.

Attention Mechanism. Given the former illustration on the limitations of unidirectional attention in typical causal LLMs, we emphasize it restricts the visibility of each token to the entire context. Hence, unlike previous works that rely on non-fundamental solutions like prompt engineering, we address this issue by directly modifying attention to be bidirectional during fine-tuning, which makes prompt design easier and inference more efficient.

3.2.2 Representation Generation

Following Wang et al. (2023) and Li et al. (2024), given a raw query-passage pair (q, p) for a specific embedding task, we first construct the instructed query input text as

$$q_{\text{ins}} = \langle \text{Instruct} \rangle \{\text{task\_definition}\} \langle \text{query} \rangle \{q\}. \tag{3}$$

Here *task_definition* refers to the definition of the specific embedding task, guiding the model to adapt towards that task. On the other hand, the input of the passage part is solely the original text. Following Wang et al. (2023) and Li et al. (2024), a [EOS] token is also appended to the end of the sequence.

We then feed such an input into the modified LLM, and derive a series of logits vectors $L = (l_1, l_2, \ldots, l_n), l_i \in \mathbb{R}^k$, where n is the sequence length and k is our clustering size. To obtain the final embeddings, following Formal et al. (2021a), log-saturation and max-pooling will be applied to L along the sequence dimension which is similar to Eq. 1, and 2.

It is also worth noting that we only employ tokens corresponding to the original query q to derive the output of the query, avoiding the noise brought by *task_definition* tokens, inspired by BehnamGhader et al. (2024). Moreover, considering the autoregressive nature of LLMs that each logit is used for the prediction of the subsequent position, we shift the logits during pooling. In other words, we regard the logit corresponding to the neighboring on the left of each token as its feature during computation.

3.2.3 Training

Recent research has explored various complex methods for training embedding models. For example, NV-Embed-v2 (Lee et al., 2024a) employs a two-stage training pipeline while also incorporating positive-aware hard-negative mining and synthetic data generation. In contrast, for simplicity and fair comparison, the training of LENS strictly adheres to the training procedure of BGE-en-ICL (Li et al., 2024), an SOTA LLM-based dense embedding model. It uses a single-stage training process and relies exclusively on publicly available data.

Given a processed input pair (q_{ins}, p), we utilize the InfoNCE loss as our objective,

$$\mathcal{L} = -\log \frac{\exp(\text{sim}(q_{\text{ins}}, p)/\tau)}{\exp(\frac{\text{sim}(q_{\text{ins}}, p)}{\tau}) + \sum_{j=1}^{N} \exp(\frac{\text{sim}(q_{\text{ins}}, p_j^-)}{\tau})} \tag{4}$$

Here p_j^- and N denote the negative passage and number of negative passages, respectively. $\text{sim}()$ is the cosine similarity function, defined as $\text{sim}() = \cos(h_{q_{\text{ins}}}, h_p)$, where $h_{q_{\text{ins}}} \in \mathbb{R}^k$ and $h_p \in \mathbb{R}^k$ are the lexicon-based embeddings from the LLM for the instructed query and passage. The temperature τ is set to 0.02 in our experiments.

4 Experiments

4.1 Setups

To ensure a fair comparison between dense embeddings and LENS, we strictly adhere to the training recipe of the SOTA dense model, BGE-en-ICL.

Model Setup. The Mistral-7B-v0.1 (Jiang et al., 2023) model is used as the backbone in LENS, in line with recent works such as BGE-en-ICL (Li et al., 2024), E5-Mistral (Wang et al., 2023), NV-Embed-v2 (Lee et al., 2024a), and LLM2Vec (BehnamGhader et al., 2024). To investigate the effect of different clustering sizes to consolidate the output token embeddings, we set k in KMeans clustering to 4,000 and 8,000 clusters, referred to as LENS-4000 and LENS-8000, respectively. LENS-4000 can output 4000-d embeddings, which is comparable to the 4096-d dense embeddings produced by the same backbone LLM.

Training Data. We directly utilize the publicly available training data provided by BGE-en-ICL. This dataset is a mixture of retrieval, reranking, clustering, classification, and semantic textual similarity (STS) tasks. Details about the training data can be found in Appendix A.2. We use the same set of task instructions as BGE-en-ICL, refer to Appendix A.3 for details.

Training Configurations. Following BGE-en-ICL, our model is trained for one epoch using LoRA (Hu et al., 2021), where the LoRA rank is 32 and the alpha is 64, and the learning rate is set to 1e-4. Each training sample is composed of 1 positive and 7 hard negatives. For retrieval tasks, we use a batch size of 512, whereas a batch size of 256 is used for the rest tasks. All data are drawn from the same dataset within the same batch. In retrieval tasks, we employ in-batch negatives and apply a KL-divergence loss to distill ranking scores from the BGE-reranker model. The maximum length for both the query and passage is set to 512. It should be noted that we deviate from BGE-en-ICL by omitting in-context learning samples during training and concentrate on zero-shot scenarios solely. It enables us to exclusively evaluate LENS performance, free from extraneous signals.

Evaluations. We evaluate the performance of various embedding models using MTEB (Muennighoff et al., 2023) and AIR-Bench (Zeng et al., 2024). MTEB is a comprehensive text embedding benchmark encompassing seven task types across a total of 56 datasets. AIR-Bench, on the other hand, spans diverse domains for retrieval tasks, including law, healthcare, and books, having no overlap with MTEB. Notably, the ground truth for the test set in AIR-Bench is hidden, and we use the 24.04 version to assess the model's out-of-domain capabilities.

We compare LENS to numerous baselines, including E5-mistral-7b-instruct (Wang et al., 2023), NV-Embed-v1/v2 (Lee et al., 2024a), gte-Qwen2-7B-instruct (Li et al., 2023b),

Task	#Dims	Retr.	Rerank.	Clust.	PairClass.	Class.	STS	Summ.	Avg.
# of datasets →		15	4	11	3	12	10	1	56
Non-Fully Public Training Data									
E5-mistral-7b-instruct	4096	56.90	60.21	50.26	88.34	78.47	84.66	31.40	66.63
Linq-Embed-Mistral	4096	60.19	60.29	51.42	88.35	80.20	84.97	30.98	68.17
voyage-large-2-instruct	1024	58.28	60.09	53.35	89.24	81.49	84.31	30.84	68.23
stella_en_400M_v5	8192	58.97	60.16	56.70	87.74	86.67	84.22	31.66	70.11
gte-Qwen2-7B-instruct	3584	60.25	61.42	56.92	85.79	86.58	83.04	31.35	70.24
SFR-Embedding-2_R	4096	60.18	60.14	56.17	88.07	89.05	81.26	30.71	70.31
stella_en_1.5B_v5	8192	61.01	61.21	57.69	88.07	87.63	84.51	31.49	71.19
NV-Embed-v2	4096	62.65	60.65	58.46	88.67	90.37	84.31	30.70	72.31
Fully Public Training Data									
LLM2Vec-Mistral-supervised	4096	55.99	58.42	45.54	**87.99**	76.63	84.09	29.96	64.80
GritLM-7B	4096	57.41	60.49	50.61	87.16	79.46	83.35	30.37	66.76
NV-Embed-v1	4096	59.36	60.59	52.80	86.91	87.35	82.84	**31.20**	69.32
bge-multilingual-gemma2	3584	59.24	59.72	54.65	85.84	88.08	83.88	**31.20**	69.88
BGE-en-ICL (zero-shot)	4096	61.67	59.66	57.51	86.93	**88.62**	83.74	30.75	71.24
LENS-4000 (***Ours***)	4000	60.76	60.86	57.92	87.93	88.13	84.35	31.56	71.21
LENS-8000 (***Ours***)	8000	**61.86**	**60.91**	**58.02**	87.98	88.43	**84.67**	29.54	**71.62**

Table 1: Top-performing models on the MTEB leaderboard as of December 1, 2024 compared to LENS. #Dims refers to the embedding dimensions. Abbreviations: Retr. = Retrieval; Rerank. = Reranking; Clust. = Clustering; PairClass. = Pair Classification; Class. = Classification; STS = Semantic Textual Similarity; Summ. = Summarization. The best and the second best results using public data are in **bold** and underlined font respectively.

Domain	#Dims	wiki	web	news	healthcare	law	finance	arxiv	msmarco	Avg.
# of datasets →		1	1	1	1	1	1	1	1	8
E5-mistral-7b-instruct	4096	61.67	44.41	48.18	56.32	19.32	54.79	44.78	59.03	48.56
Linq-Embed-Mistral	4096	61.04	48.41	49.44	**60.18**	20.34	50.04	47.56	60.50	49.69
NV-Embed-v1	4096	62.84	50.42	51.46	58.53	20.65	49.89	46.10	60.27	50.02
gte-Qwen2-7B-instruct	3584	63.46	51.20	54.07	54.20	22.31	**58.20**	40.27	58.39	50.26
stella_en_1.5B_v5	8192	61.99	50.88	53.87	58.81	23.22	57.26	44.81	61.38	51.53
SFR-Embedding-Mistral	4096	63.46	51.27	52.21	58.76	23.27	56.94	47.75	58.99	51.58
NV-Embed-v2	4096	65.19	52.58	53.13	59.56	25.00	53.04	**48.94**	60.80	52.28
BGE-en-ICL (zero-shot)	4096	64.61	54.40	55.11	57.25	25.10	54.81	48.46	**63.71**	**52.93**
LENS-4000 (***Ours***)	4000	62.60	52.06	52.49	57.23	24.08	48.87	43.78	61.17	50.28
LENS-8000 (***Ours***)	8000	**65.50**	**54.52**	**55.16**	58.20	**25.62**	54.57	45.45	63.00	52.75

Table 2: QA performance on AIR-Bench 24.04 (English) across different models, where nDCG@10 is used as the metric. #Dims refers to the embedding dimensions. The best and the second best results across all models are in **bold** and underlined font respectively.

LLM2Vec (BehnamGhader et al., 2024), SFR-Embedding-2_R (Meng et al., 2024), GritLM-7B (Muennighoff et al., 2024), and BGE-en-ICL (Li et al., 2024). The results of PromptReps and Mistral-Splade are excluded, as they are designed specifically for retrieval tasks and their performance falls below of the weakest baseline, namely LLM2Vec-Mistral-supervised. Some of the baselines use private data during training or involve in-context learning. To ensure a fair comparison, we focus on **zero-shot scenarios** where no few-shot sample is included in the prompt, e.g., BGE-en-ICL.

4.2 Main results

MTEB. Table 1 demonstrates the results of a variety of models on MTEB. LENS-8000 achieves the highest average performance among all models trained on fully public data as of December 1, 2024. Notably, LENS-8000 outperforms BGE-en-ICL, its dense embedding counterpart trained with the same data and hyperparameters. 6 among 7 categories of tasks also demonstrate consistent superiority. Besides, LENS-4000 yields comparable performance as BGE-en-ICL, both share equivalent feature dimensions, but our lexicon-based method can deliver better transparency. Furthermore, LENS-8000 ranks second among all models in overall average performance. The leading model, NV-Embed-v2, attains its superiority through a significantly more complex training pipeline, which includes a two-stage training pipeline, positive-aware hard-negative mining, and synthetic data generation. By contrast, LENS uses fully public data and

Text	Top-weighted clusters
most dependable affordable cars	(cars, Cars), (cheap, affordable), (reliable, reli), (depend, depends), (aff, afford)
fastest growing bonsai trees	(faster, fastest), (grow, growing), (fast, Fast), (tree, trees), (quickly, rapid)
causes of hypoxia in adults	(adult, adults), (oxygen, oxy), (cause, caused), (hyp, yp), (ox, Ox)
weather in lisbon april	(Portug, Portuguese), (bon, Bon), (weather, rather), (Spring, spring), (AP, #AP)
other hot flashes causes	(hot, Hot), (cause, causes), (flash, Flash), (flush, #flush), (heat, Heat)

Table 3: Qualitive examples of LENS-8000. For each example, the top-5 clusters with the largest weights in the embeddings are shown, with two tokens from each cluster included.

Clusters
quickly, rapid, rapidly, swift
cannot, impossible, Unable, Cannot, Unable
shows, shown, showed, showing
review, Review, reviews, reviewed, Reviews
educ, education, Educ, Education, educational, Edu

Table 4: Cluster examples of LENS-8000. Each row presents tokens belonging to a single cluster. More cluster examples are provided in Appendix A.1.

adopts a simpler training procedure.

AIR-Bench. We also evaluate LENS along with baselines on the QA tasks of AIR-Bench. As shown in Table 2, LENS-8000 outperforms the top-performing model on MTEB, NV-Embed-v2, demonstrating its promising generalization capabilities. Despite slightly lagging behind its dense counterpart BGE-en-ICL, LENS still remains competitive in several sub-tasks. However, LENS-4000 performs less competitively, potentially because a smaller number of clusters may result in overgeneralized clusters and information loss.

4.3 Qualitative Examples

We present some clustering results in Table 4. It can be found that: i) LENS groups semantically equivalent tokens (e.g., rapid and quickly, cannot and impossible); ii) it groups morphologically similar tokens (e.g., shows and showed); and iii) group uppercase/lowercase variants and whole-word/subword forms (e.g., review and Review, Edu and education). Such an observation proves the effect of clustering to eliminate the redundancy and noise of the tokenizer in some ways as we expected.

In addition, qualitative examples from MS MARCO of LENS-8000 embeddings are given in Table 3. The top-5 clusters with the largest weights in the embeddings are presented for each sample, where two tokens from each cluster are included. Obviously, these clusters are highly semantically relevant to the input texts, which can be regarded as some keywords. There are also interesting findings that the embeddings show some deep understanding of the text such as oxygen in response to the input "causes of hypoxia in adults", and some knowledge expansion capabilities like Portuguese and spring for the input "weather in lisbon april". These qualitative samples demonstrate that lexicon-based embeddings from LLMs captures more contextual features rather than some shallow token meanings.

Figure 3: Influence of the number of clusters. The configuration with 32,000 clusters retains the original token embeddings without clustering.

5 Analysis

In this section, we conduct a detailed investigation of LENS. For the sake of computational resources, we reduce the training data of each dataset to 10% of its original size in this part. Besides, we use the same MTEB subset as Jiang et al. (2024) for faster evaluation, as it correlates well with the overall performance of MTEB (details in Appendix A.4).

5.1 Influence of the Number of Clusters

We investigate how the number of clusters k affects the performance, as shown in Figure 3. The configuration with 32,000 clusters retains the original token embeddings without applying clustering. Our analysis reveals that decreasing the number of output entries from 32,000 tokens consistently improves performance, even when the number of clusters is reduced to as low as 2,000. However, it is crucial to maintain an adequate number of clusters to prevent information loss that can arise from overgeneralization. A configuration of 8,000 clusters strikes a good balance between effectiveness

Task	Retr.	Rerank.	Clust.	PairClass.	Class.	STS	Summ.	Avg.
# of datasets →	1	1	1	1	1	1	1	7
Unidirectional Attention								
Last-token pooling	73.84	65.19	60.46	96.69	58.66	89.26	30.05	67.73
Sum-pooling	72.46	59.57	50.55	89.90	54.64	80.55	29.70	62.48
Max-pooling	75.18	59.68	50.93	92.06	57.58	82.74	30.89	64.15
Bidirectional Attention								
Last-token pooling	76.89	64.21	61.57	96.62	58.33	88.72	30.72	68.15
Sum-pooling	75.65	63.64	61.77	96.97	60.05	89.58	30.98	68.38
Max-pooling	76.19	64.53	63.05	97.03	62.30	88.92	31.49	69.07

Table 5: Influence of attention mechanisms and pooling methods.

Dataset	ARG	CLI	CQA	DBP	FEV	FIQ	HOT	MSM	NFC	NQ	QUO	SCD	SCF	TOU	COV	Avg.
BGE-en-ICL	82.76	45.35	47.23	50.42	91.96	58.77	84.98	46.72	40.69	73.85	91.02	25.25	78.33	29.67	78.11	61.67
LENS-8000 (***Ours***)	76.02	45.77	48.67	49.75	92.32	61.57	85.71	47.24	40.61	74.64	90.79	28.54	79.75	29.34	77.18	61.86
NV-Embed-v2	70.07	45.39	50.24	53.50	93.75	65.73	85.48	45.63	45.17	73.57	89.04	21.90	80.13	31.78	88.44	62.65
LENS (***Ours***) + BGE	81.37	**47.14**	48.57	**51.79**	**93.12**	**62.00**	**87.12**	**47.66**	**41.55**	**75.81**	**91.07**	28.41	**80.19**	**30.51**	**78.72**	**63.00**

Table 6: Results in terms of nDCG@10 on the retrieval subset (i.e. BEIR) of MTEB. We use the first three letters of each dataset's name as its abbreviation, except SCIDOCS (abbreviated as SCD) and SciFact (abbreviated as SCF). **Bolded values** indicate datasets where the combinations outperform both LENS-8000 and BGE-en-ICL individually. On 12 out of 15 datasets, combining LENS-8000 and BGE-en-ICL results in improved performance.

and efficiency (dimensionality). Consequently, we employ $k = 8,000$ in the subsequent experiments.

5.2 Influence of Model Architcure

We extensively investigate the effects of the attention mechanism and pooling methods, as illustrated in Table 5. For attention, we examine both unidirectional and bidirectional attention. Regarding pooling strategies, we assess max-pooling, sum-pooling, and last-token pooling. The results highlight the critical role of bidirectional attention in achieving strong performance with lexicon-based embeddings, as evidenced by its superiority across all pooling methods. Among these pooling methods, max-pooling emerges as the most effective strategy. This finding partially explains the poor performance of lexicon-based embeddings from PromptReps, which relies on last-token pooling with unidirectional attention.

5.3 Hybrid Lexicon-Dense Embeddings

Previous studies have demonstrated that lexicon-based embeddings and dense embeddings are complementary, and combining them can lead to significant performance improvements. In this section, we explore the effectiveness of combining LENS with BGE-en-ICL, both trained on the same data but representing different types of embeddings. To evaluate general-use cases, we concatenate the two embeddings into a single embedding, without applying any additional operations. We hypothesize that enhanced performance could be achieved by tuning the combination weights of the two embeddings.

The results are presented in Table 6. Combining LENS-8000 with BGE-en-ICL yields a substantial performance improvement, increasing from 61.67/61.86 to 63.00, which surpasses NV-Embed-v2 and achieves SOTA results on the retrieval subset of MTEB as of December 1, 2024. Furthermore, such an improvement is consistent, as evidenced by performance gains on 12 out of 15 datasets.

6 Conclusion

In this work, we introduce LENS, a simple yet effective framework for generating lexicon-based text embeddings using LLMs. Our approach leverages token embedding clustering to address the redundancy challenges inherent in LLM tokenizers, while also enabling bidirectional attention to fully unlock the potential of LLMs. Extensive experiments demonstrate the promising effectiveness and generalization capabilities of LENS compared to SOTA dense embeddings. Qualitative examples reveal that LENS produces embeddings that are grounded and demonstrate a deep understanding of the input. Further analyses show the superiority of fusing lexicon-based LENS and dense embeddings, which surpasses each individual model on the retrieval subset of MTEB (i.e., BEIR).

Limitations

We acknowledge the following limitations of our work. First, our training and evaluation are limited to English, leaving multilingual datasets, such as Miracl (Zhang et al., 2023), unexplored. This restricts the generalizability of our findings to non-English contexts. Second, we applied LENS exclusively to the widely used Mistral-7B model, leaving other models unexplored. Additionally, compared to previous lexicon-based models like SPLADE, utilizing LLMs as the backbone significantly increases computational costs.

References

C.J. Adams, Daniel Borkan, Jeffrey Sorensen, Lucas Dixon, Lucy Vasserman, and Nithum Thain. 2019. Jigsaw unintended bias in toxicity classification.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. in* sem 2012: The first joint conference on lexical and computational semantics–volume 1: Proceedings of the main conference and the shared task, and volume 2: Proceedings of the sixth international workshop on semantic evaluation (semeval 2012). *Association for Computational Linguistics.*

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. *arXiv 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. *arXiv preprint arXiv:2404.05961*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Xi Chen, Ali Zeynali, Chico Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw Grabowicz, Scott Hale, David Jurgens, and Mattia Samory. 2022. Semeval-2022 task 8: Multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1094–1106.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

DataCanary, hilfialkaff, Lili Jiang, Meg Risdal, Nikhil Dandekar, and tomtung. 2017. Quora question pairs.

Hervé Déjean, Stephane Clinchant, Carlos Lassance, Simon Lupart, and Thibault Formal. 2023. Benchmarking middle-trained language models for neural search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23.

Meet Doshi, Vishwajeet Kumar, Rudra Murthy, Vignesh P, and Jaydeep Sen. 2024. Mistral-splade: Llms for better learned sparse retrieval. *arXiv preprint arXiv:2110.01529*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2024. Towards effective and efficient sparse neural information retrieval. *ACM Trans. Inf. Syst.*

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021a. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021b. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21.

John A Hartigan and Manchek A Wong. 1979. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108.

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*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q. Jiang, Alicja Ziarko, Bartosz Piotrowski, Wenda Li, Mateja Jamnik, and Piotr Miłoś. 2024. Repurposing language models into embedding models: Finding the compute-optimal recipe. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine learning proceedings 1995*, pages 331–339. Elsevier.

Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. Splade-v3: New baselines for splade. *arXiv preprint arXiv:2403.06789*.

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

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. 2024b. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327*.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023a. Making large language models a better foundation for dense retrieval. *arXiv preprint arXiv:2312.15503*.

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*.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962.

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

Jimmy Lin. 2021. A proposed conceptual framework for a representational approach to information retrieval. *arXiv preprint arXiv:2110.01529*.

Jimmy Lin and Xueguang Ma. 2021. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. *arXiv preprint arXiv:2106.14807*.

Xueqing Liu, Chi Wang, Yue Leng, and ChengXiang Zhai. 2018. Linkso: a dataset for learning to retrieve similar question answer pairs on software development forums. In *Proceedings of the 4th ACM SIGSOFT International Workshop on NLP for Software Engineering*, pages 2–5.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Wei Chen Maggie, Phil Culliton. 2020. Tweet sentiment extraction.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: financial opinion mining and question answering. In *Companion proceedings of the the web conference 2018*, pages 1941–1942.

Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonellotto. 2021. Learning passage impacts for inverted indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 1723–1727.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Sfr-embedding-2: Advanced text embedding with multi-stage training.

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

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.

Thong Nguyen, Sean MacAvaney, and Andrew Yates. 2023. A unified framework for learned sparse retrieval. In *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part III*, pages 101–116. Springer.

James O'Neill, Polina Rozenshtein, Ryuichi Kiryo, Motoko Kubota, and Danushka Bollegala. 2021. I wish i would have loved this one, but i didn't–a multilingual dataset for counterfactual detection in product review. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7092–7108.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. Carer: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3687–3697.

Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Xiaolong Huang, Binxing Jiao, Linjun Yang, and Daxin Jiang. 2023a. LexMAE: Lexicon-bottlenecked pretraining for large-scale retrieval. In *The Eleventh International Conference on Learning Representations*.

Tao Shen, Xiubo Geng, Chongyang Tao, Can Xu, Guodong Long, Kai Zhang, and Daxin Jiang. 2023b. Unifier: A unified retriever for large-scale retrieval. KDD '23.

Aina Garí Soler, Matthieu Labeau, and Chloé Clavel. 2024. The impact of word splitting on the semantic content of contextualized word representations. *Transactions of the Association for Computational Linguistics*.

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

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Yi Zeng, Yu Yang, Andy Zhou, Jeffrey Ziwei Tan, Yuheng Tu, Yifan Mai, Kevin Klyman, Minzhou Pan, Ruoxi Jia, Dawn Song, et al. 2024. Airbench 2024: A safety benchmark based on risk categories from regulations and policies. *arXiv preprint arXiv:2407.17436*.

Xinyu Zhang, Jing Lu, Vinh Q. Tran, Tal Schuster, Donald Metzler, and Jimmy Lin. 2024. Tomato, tomahto, tomate: Measuring the role of shared semantics among subwords in multilingual language models. *arXiv preprint arXiv:2411.04530*.

Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. 2023. Miracl: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131.

Shengyao Zhuang, Xueguang Ma, Bevan Koopman, Jimmy Lin, and Guido Zuccon. 2024. PromptReps: Prompting large language models to generate dense and sparse representations for zero-shot document retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Shengyao Zhuang and Guido Zuccon. 2021. Fast passage re-ranking with contextualized exact term matching and efficient passage expansion. *arXiv preprint arXiv:2108.08513*.

A Appendix

A.1 Clustering results

Clusters
`impact, Impact, impacts`
`Entity, entity, #Entity, Entities, entities`
`TV, television, tv, Television, televis`
`comfort, comfortable, comfort`
`beautiful, lovely, gorgeous, handsome, beautifully`
`guy, guys, Guy, dude`
`fit, FIT, fits, fitting, fitted`
`recomm, recommend, recommended, recommendation`
`star, stars, Stars`
`reach, reached, reaching, reaches, reach`

Table 7: Cluster examples of LENS-8000. Each row presents tokens belonging to a single cluster.

A.2 Training data details

We leverage the public training data provided by BGE-en-ICL (Li et al., 2024). Specifically, the training data is a mixture of retrieval, reranking, classification, clustering, and STS data.

- **Retrieval**: ELI5 (Fan et al., 2019), HotpotQA (Yang et al., 2018), FEVER (Thorne et al., 2018), MSMARCO passage and document ranking (Bajaj et al., 2018), NQ, NLI, SQuAD, TriviaQA, Quora Duplicate Questions (DataCanary et al., 2017), Arguana (Wachsmuth et al., 2018), FiQA (Maia et al., 2018).

- **Reranking**: SciDocsRR (Cohan et al., 2020), StackOverFlowDupQuestions (Liu et al., 2018).

- **Classification**: AmazonReviews-Classification (McAuley and Leskovec, 2013), AmazonCounterfactual-Classification (O'Neill et al., 2021), Banking77-Classification (Casanueva et al., 2020), Emotion-Classification (Saravia et al., 2018), TweetSentimentExtraction-Classification (Maggie, 2020), MTOPIntent-Classification (Li et al., 2021), IMDB-Classification (Maas et al., 2011), ToxicConversations-Classification (Adams et al., 2019).

- **Clustering**: TwentyNewsgroups-Clustering (Lang, 1995), {Arxiv/Biorxiv/Medrxiv/Reddit/StackExchange}-Clustering-{S2S/P2P}

- **STS**: STS12 (Agirre et al., 2012), STS22 (Chen et al., 2022), STS-Benchmark (Cer et al., 2017).

A.3 Task Instructions

We present the task instructions we used in Table 8.

A.4 MTEB Subset Details

Following Jiang et al. (2024), for each task category, we select one dataset for evaluation. The chosen dataset is determined based on the model results presented in the original MTEB paper, focusing on the dataset with the highest correlation to the category's average performance.

- **Classification**: EmotionClassification
- **Clustering**: TwentyNewsgroupsClustering
- **Pair classification**: SprintDuplicateQuestions
- **Reranking**: AskUbuntuDupQuestions
- **Retrieval**: SciFact
- **Semantic text similarity**: STS15
- **Summarization**: SummEval

A.5 Detailed MTEB Results

We present the detailed MTEB results in Table 9.

Task Name	Instruction
ArguAna	Given a claim, find documents that refute the claim.
ClimateFEVER	Given a claim about climate change, retrieve documents that support or refute the claim.
CQADupStack	Given a question, retrieve detailed question descriptions from Stackexchange that are duplicates to the given question.
DBPedia	Given a query, retrieve relevant entity descriptions from DBPedia.
FEVER	Given a claim, retrieve documents that support or refute the claim.
FiQA2018	Given a financial question, retrieve user replies that best answer the question.
HotpotQA	Given a multi-hop question, retrieve documents that can help answer the question.
MSMARCO	Given a web search query, retrieve relevant passages that answer the query.
NFCorpus	Given a question, retrieve relevant documents that best answer the question.
Natural Question	Given a question, retrieve Wikipedia passages that answer the question.
QuoraRetrieval	Given a question, retrieve questions that are semantically equivalent to the given question.
SCIDOCS	Given a scientific paper title, retrieve paper abstracts that are cited by the given paper.
SciFact	Given a scientific claim, retrieve documents that support or refute the claim.
Touche2020	Given a question, retrieve detailed and persuasive arguments that answer the question.
TREC-COVID	Given a query, retrieve documents that answer the query.
STS*	Retrieve semantically similar text.
SummEval	Given a news summary, retrieve other semantically similar summaries.
AmazonCounterfactualClassification	Classify a given Amazon customer review text as either counterfactual or not-counterfactual.
AmazonPolarityClassification	Classify Amazon reviews into positive or negative sentiment.
AmazonReviewsClassification	Classify the given Amazon review into its appropriate rating category.
Banking77Classification	Given a online banking query, find the corresponding intents.
EmotionClassification	Classify the emotion expressed in the given Twitter message into one of the six emotions: anger, fear, joy, love, sadness, and surprise.
ImdbClassification	Classify the sentiment expressed in the given movie review text from the IMDB dataset.
MassiveIntentClassification	Given a user utterance as query, find the user intents.
MassiveScenarioClassification	Given a user utterance as query, find the user scenarios.
MTOPDomainClassification	Classify the intent domain of the given utterance in task-oriented conversation.
MTOPIntentClassification	Classify the intent of the given utterance in task-oriented conversation.
ToxicConversationsClassification	Classify the given comments as either toxic or not toxic.
TweetSentimentExtractionClassification	Classify the sentiment of a given tweet as either positive, negative, or neutral.
ArxivClusteringP2P	Identify the main and secondary category of Arxiv papers based on the titles and abstracts.
ArxivClusteringS2S	Identify the main and secondary category of Arxiv papers based on the titles.
BiorxivClusteringP2P	Identify the main category of Biorxiv papers based on the titles and abstracts.
BiorxivClusteringS2S	Identify the main category of Biorxiv papers based on the titles.
MedrxivClusteringP2P	Identify the main category of Medrxiv papers based on the titles and abstracts.
MedrxivClusteringS2S	Identify the main category of Medrxiv papers based on the titles.
RedditClustering	Identify the topic or theme of Reddit posts based on the titles.
RedditClusteringP2P	Identify the topic or theme of Reddit posts based on the titles and posts.
StackExchangeClustering	Identify the topic or theme of StackExchange posts based on the titles.
StackExchangeClusteringP2P	Identify the topic or theme of StackExchange posts based on the given paragraphs.
TwentyNewsgroupsClustering	Identify the topic or theme of the given news articles.
AskUbuntuDupQuestions	Retrieve duplicate questions from AskUbuntu forum.
MindSmallReranking	Retrieve relevant news articles based on user browsing history.
SciDocsRR	Given a title of a scientific paper, retrieve the titles of other relevant papers.
StackOverflowDupQuestions	Retrieve duplicate questions from StackOverflow forum.
SprintDuplicateQuestions	Retrieve duplicate questions from Sprint forum.
TwitterSemEval2015	Retrieve tweets that are semantically similar to the given tweet.
TwitterURLCorpus	Retrieve tweets that are semantically similar to the given tweet.
AIR-Bench	Given a question, retrieve passages that answer the question.

Table 8: Task instructions for MTEB and AIR-Bench benchmarks.

Dataset	gte-Qwen2- 7B-instruct	SFR-Embe dding-2_R	stella_en_ 1.5B_v5	BGE-en-ICL (zero-shot)	NV-Em bed-v2	LENS -4000	LENS -8000
ArguAna	64.27	62.34	65.27	82.76	70.07	77.32	76.02
ClimateFEVER	45.88	34.43	46.11	45.35	45.39	44.62	45.77
CQADupStack	46.43	46.11	47.75	47.23	50.24	47.39	48.67
DBPEDIA	52.42	51.21	52.28	50.42	53.50	50.10	49.75
FEVER	95.11	92.16	94.83	91.96	93.75	92.37	92.32
FiQA2018	62.03	61.77	60.48	58.77	65.73	60.43	61.57
HotpotQA	73.08	81.36	76.67	84.98	85.48	85.07	85.71
MSMARCO	45.98	42.18	45.22	46.72	45.63	46.95	47.24
NFCorpus	40.60	41.34	42.00	40.69	45.17	41.64	40.61
Natural Question	67.00	73.96	71.80	73.85	73.57	73.13	74.64
QuoraRetrieval	90.09	89.58	90.03	91.02	89.04	90.84	90.79
SCIDOCS	28.91	24.87	26.64	25.25	21.90	27.51	28.54
SciFact	79.06	85.91	80.09	78.33	80.13	78.39	79.75
Touche2020	30.57	28.18	29.94	29.67	31.78	25.86	29.34
TREC-COVID	82.26	87.28	85.98	78.11	88.44	69.73	77.18
BIOSSES	81.37	87.60	83.11	86.35	87.42	84.47	85.83
SICK-R	79.28	77.01	82.89	83.87	82.15	83.81	83.30
STS12	79.55	75.67	80.09	77.73	77.89	79.07	80.99
STS13	88.83	82.40	89.68	85.98	88.30	86.54	87.34
STS14	83.87	79.93	85.07	82.34	84.30	84.32	84.39
STS15	88.54	85.82	89.39	87.35	89.04	89.69	89.75
STS16	86.49	84.50	87.15	86.54	86.77	87.23	87.63
STS17	88.73	88.93	91.35	91.25	90.67	91.55	90.87
STS22	66.88	67.10	68.10	68.08	68.12	68.69	68.09
STSBenchmark	86.85	83.60	88.23	87.92	88.41	88.22	88.47
SummEval	31.35	30.71	31.49	30.75	30.70	31.55	29.54
SprintDuplicateQuestions	92.82	97.62	96.04	95.06	97.02	96.98	97.00
TwitterSemEval2015	77.96	78.57	80.58	78.54	81.11	79.31	79.56
TwitterURLCorpus	86.59	88.03	87.58	87.19	87.87	87.50	87.37
AmazonCounterfactual	91.31	92.72	92.87	92.88	94.28	93.61	93.69
AmazonPolarity	97.50	97.31	97.16	96.86	97.74	97.05	97.07
AmazonReviews	62.56	61.04	59.36	61.28	63.96	62.83	63.61
Banking77	87.57	90.02	89.79	91.42	92.42	90.43	90.19
Emotion	79.45	93.37	84.29	93.31	93.38	92.33	91.87
Imdb	96.75	96.80	96.66	96.91	97.14	97.12	97.00
MassiveIntent	85.41	85.97	85.83	82.26	86.10	79.65	81.14
MassiveScenario	89.77	90.61	90.20	83.92	92.17	81.97	83.53
MTOPDomain	99.04	98.58	99.01	97.99	99.25	97.49	97.44
MTOPIntent	91.88	91.30	92.78	93.56	94.37	92.59	92.81
ToxicConversations	85.12	91.14	88.76	93.16	92.74	92.29	92.37
TweetSentimentExtraction	72.58	79.70	74.84	79.90	80.87	80.17	80.42
Arxiv-P2P	54.46	54.02	55.44	54.42	55.80	54.87	54.81
Arxiv-S2S	51.74	48.82	50.66	49.17	51.26	50.25	50.14
Biorxiv-P2P	50.09	50.76	50.68	52.32	54.09	52.39	52.48
Biorxiv-S2S	46.65	46.57	46.87	48.38	49.60	48.35	48.52
Medrxiv-P2P	46.23	46.66	46.87	46.13	46.09	46.35	46.38
Medrxiv-S2S	44.13	44.18	44.65	44.20	44.86	44.54	44.89
Reddit	73.55	62.92	72.86	71.20	71.10	72.32	72.37
Reddit-P2P	74.13	72.74	75.27	72.17	74.94	73.20	73.89
StackExchange	79.86	76.48	80.29	81.29	82.10	81.70	81.60
StackExchange-P2P	49.41	48.29	49.57	45.53	48.36	43.73	44.41
TwentyNewsgroups	53.91	66.42	61.43	68.51	64.82	69.44	68.78
AskUbuntuDupQuestions	67.58	66.71	67.33	64.80	67.46	65.45	65.74
MindSmallRerank	33.36	31.26	33.05	30.60	31.76	31.92	31.46
SciDocsRR	89.09	87.29	89.20	86.90	87.59	87.92	87.63
StackOverflowDupQuestions	55.66	55.32	55.25	56.32	55.79	58.15	58.79
MTEB Average (56)	70.24	70.31	71.19	71.24	72.31	71.21	71.62

Table 9: Detailed MTEB results.