Conan-Embedding-v2: Training an LLM from Scratch for Text Embeddings

Shiyu Li[1], Yang Tang[1], Ruijie Liu[1], Shi-Zhe Chen[1], Xi Chen[1]*,

[1]Basic Algorithm Center, PCG, Tencent
{shyuli, ethanntang, jackrjliu, shizhechen, jasonxchen}@tencent.com

Abstract

Large language models (LLMs) have recently demonstrated excellent performance in text embedding tasks. Previous work usually use LoRA to fine-tune existing LLMs, which are limited by the data and training gap between LLMs and embedding models. In this work, we introduce Conan-embedding-v2, a new 1.4B-parameter LLM trained from scratch and fine-tuned as a text embedder. First, we add news data and multilingual pairs for LLM pretraining to bridge the data gap. Based on this, we propose a cross-lingual retrieval dataset that enables the LLM to better integrate embeddings across different languages. Second, whereas LLMs use a causal mask with token-level loss, embedding models use a bidirectional mask with sentence-level loss. This training gap makes full fine-tuning less effective than LoRA. We introduce a soft-masking mechanism to gradually transition between these two types of masks, enabling the model to learn more comprehensive representations. Based on this, we propose a dynamic hard negative mining method that exposes the model to more difficult negative examples throughout the training process. Being intuitive and effective, with only approximately 1.4B parameters, Conan-embedding-v2 achieves SOTA performance on both the Massive Text Embedding Benchmark (MTEB) and Chinese MTEB (May 19, 2025).

1 Introduction

Text embedding maps words, sentences, or documents into a high-dimensional continuous space, allowing similar texts to have closer vector representations (Mikolov et al., 2013; Karpukhin et al., 2020). This representation not only elevates the operability of text data, but also significantly improves performance in various downstream tasks (Devlin et al., 2018; Radford, 2018; Reimers, 2019). With the rapid development of large language models, LLM-based embedding models (Wang et al., 2023; Li et al., 2023; Wang et al., 2024a) have played a crucial role in text representation and information retrieval tasks.

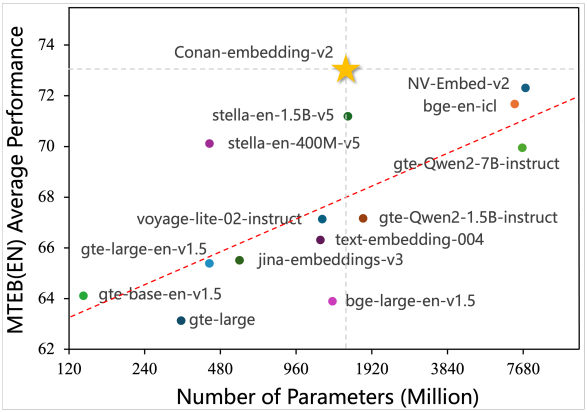


Figure 1: **Comparison between Conan-embedding-v2 and other embedding models on MTEB English benchmark (May 19, 2025)** (Muennighoff et al., 2022). This benchmark evaluates model performance across seven tasks: classification, clustering, pair classification, reranking, retrieval, semantic textual similarity, and summarization. The red dashed line depicts the logarithmic trendline fitted to the performance data of all the baseline models, excluding Conan-embedding-v2.

However, previous work with LLMs usually starts with the pretrained Mistral-7B (Jiang et al., 2023) and LoRA (Hu et al., 2021) to fine-tune the embedding models. This approach may be constrained by the disparities in the training data and process between LLMs and embedding models. First, it relies on the capabilities of the base LLMs, and there is a gap between the corpora used to train the base LLMs and the data required for embedding training. Moreover, the training paradigms for LLMs and embedding models are fundamentally different. LLMs are trained to predict the next token, whereas embedding models need to generate an embedding vector based on the entire query or candidate sentence. This training gap makes full fine-tuning less effective than LoRA, and the

*Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 15012–15028
November 4-9, 2025 ©2025 Association for Computational Linguistics

improvements through LoRA have inherent limitations (Biderman et al., 2024).

To address the above challenges, we propose Conan-embedding-v2, a new LLM trained from scratch and finetuned as a text embedder, which extends the BERT-based conan-v1 (Li et al., 2024b) in both training data and methodology. First, to bridge the data gap, Conan-embedding-v2 combines pretraining on extensive news data with fine-tuning on specialized embedding corpora during LLM training. Second, to address the training gap, we have developed a **soft mask** mechanism that facilitates a gradual transition from causal masking to bidirectional masking, allowing the rank of the mask to gradually decrease. This enables the model to learn more comprehensive feature representations during the early stages of training. Specifically, as LLMs are no longer constrained by the corpus of the backbone, we introduce a novel **cross-lingual retrieval dataset** that enables bidirectional search between 26 languages. This allows the model to integrate embeddings across diverse linguistic systems. Moreover, since LLMs are no longer constrained by LoRA, we present a **dynamic hard negative mining** that keep the high value of negative samples throughout the training process.

As shown in Figure 1, Conan-embedding-v2 demonstrates SOTA performance, outperforming both BERT-based and LLM-based methods, while maintaining an efficient model architecture with optimized size. Our key contributions can be summarized as follows:

- We propose Conan-embedding-v2, a new LLM trained from scratch and finetuned as a text embedder to address the data and training gaps between LLMs and embedding models.
- We introduce a novel cross-lingual retrieval dataset that enables bidirectional search between 26 languages, improving the integration of multilingual embeddings.
- We conduct empirical evaluations, demonstrating that our method achieves SOTA performance on both English and Chinese MTEB benchmarks, while maintaining a reasonable model size and inference speed.

2 Related Work

2.1 LLM-based Embedding Models

Recent progress in LLMs has significantly advanced the development of text embedding models, enabling more efficient and versatile representations. By fine-tuning pretrained LLMs on the synthetic data, (Wang et al., 2023) achieved outstanding performance with few training steps. The findings of this research confirmed that leveraging LLMs for embeddings proved efficient and effective. Researchers have proposed diverse approaches to enhance LLM-based text embedders from multiple perspectives. NV-Embed (Lee et al., 2024) improved representation capability through introducing latent attention layers and removing causal attention encoding. bge-en-icl (Li et al., 2024a) utilized a few-shot learning approach to generate high-quality text embeddings by taking advantage of the in-context learning ability in LLMs. NV-Retriever (Moreira et al., 2024) introduced a mining approach using positive relevance scores to eliminate false negatives, improving training efficiency and retrieval accuracy. mE5 (Wang et al., 2024a) and M3-Embedding (Chen et al., 2024b) focused on multilingual text embedding. The above research significantly improved the performance of LLM-based text embedding.

2.2 Cross-lingual Information Retrieval

While LLM-based embedding models have shown remarkable progress, their application in cross-lingual information retrieval (CLIR) presents unique challenges and opportunities (Hämmerl et al., 2024). Traditional CLIR methods struggle to support multiple languages, maintain computational efficiency, and achieve high retrieval performance simultaneously. Recent advances have demonstrated promising developments. Multilingual text embedding approaches, such as M3-Embedding (Chen et al., 2024b) and mE5 (Wang et al., 2024a), have shown remarkable capabilities in handling multiple languages while maintaining computational efficiency through contrastive learning and knowledge distillation techniques. Additionally, LECCR (Wang et al., 2024b) has begun incorporating multimodal LLMs to bridge the semantic gap between different modalities and languages, resulting in significant improvements in cross-lingual cross-modal retrieval tasks. To address the challenges of low-resource languages, recent studies (Miao et al., 2024; Litschko et al., 2024) have proposed innovative solutions using word alignment and dialect-specific approaches to enhance embedding quality.

3 Method

3.1 Overall Pipeline

Since Conan-embedding-v2 is trained from scratch, the training process is divided into four stages: LLM pre-training, LLM supervised fine-tuning (SFT), embedding weakly-supervised training, and embedding supervised training. Each stage differs in data formats and loss functions.

3.1.1 LLM Training

To better adapt large language models (LLMs) to embedding tasks, we designed Conan-embedding-v2 with 8 layers and a hidden dimension of 3584, supporting up to 32,768 input tokens. This model, totaling 1.4 billion parameters, offers a higher number of embedding dimensions with fewer parameters. We trained the Conan tokenizer on approximately 400,000 multilingual corpora, resulting in a vocabulary size of 150,000. As shown in Figure 2, we initially pre-trained the model on approximately 3T tokens of general data, with a emphasis on adding news, question-answer, and web page data. We employed the standard data filtering methods as described in (Cai et al., 2024). Following this, we collected approximately 600 million instances of SFT data using the paired data (query-positive), formatted as instruction, input, and output.

3.1.2 Embedding Training

Weakly-supervised Training. For embedding training, we first implemented weakly-supervised training to allow the model to initially learn the representations for embedding. During this stage, we use the same data as in LLM supervised fine-tuning, but with different data formats and loss functions. Specifically, we provide the instruction and input as the query, and the output as the positive passage. To ensure higher data quality, we utilize the gte-Qwen2-7B-instruct model (Li et al., 2023) for scoring and discard any data with scores below 0.4. To efficiently and effectively leverage the pair data, we employ the InfoNCE loss with In-Batch Negative sampling (Gutmann and Hyvärinen, 2010) during training, the formula is:

$$\mathcal{L}_{neg} = -\sum_{i=1}^{N} \log \frac{\exp(\cos(x_i, y_i^+))}{\sum_{j=1}^{M} \exp(\cos(x_i, y_i))} \quad (1)$$

x_i represents the query of the positive sample, y_i^+ represents the passage of the positive sample, y_i represents the passages of other samples in the batch, which are considered as negative samples.

Supervised Training. After weakly-supervised training, we perform task-specific fine-tuning for different downstream tasks. As shown in Figure 2, we divide the tasks into four categories: retrieval, cross-lingual retrieval, classification and STS (semantic textual similarity). The first three tasks include a query, a positive text, and some negative texts, utilizing the classic InfoNCE loss function. STS task involves distinguishing the similarity between two texts, with the classic loss function being cross-entropy loss. According to (Su, 2022) and other works (Wang Yuxin, 2023), CoSENT loss is slightly better than cross-entropy loss. Therefore, we also adopt CoSENT loss to optimize STS task, which is formulated as follows:

$$\mathcal{L}_{cos} = \log\left(1 + \sum_{Order} \exp \frac{\langle x_k, x_l\rangle - \langle x_i, x_j\rangle}{\tau}\right) \quad (2)$$

where $Order = \text{sim}(i,j) > \text{sim}(k,l)$, $\text{sim}(k,l)$ is the ground-truth similarity between x_i and x_j. $\langle x_k, x_l\rangle$ represents the cosine similarity between x_k and x_l. τ is the scale temperature.

3.2 Soft Mask

During the training phase of LLMs, a causal mask is employed to ensure that the current token does not have access to subsequent tokens, which is suitable for token-level language modeling. However, embedding training requires a holistic understanding of the sentence, using a bidirectional mask for vector-level modeling. These two types of masks have several key gaps.

First, since the upper triangle of the causal mask is entirely zeros, the attention weights in this region are not used during the forward propagation. When switching directly to a bidirectional mask, these weights require a learning process to become effective. Second, the causal mask is full-rank, providing stronger expressive power, whereas the rank of the bidirectional mask is always one. If we directly switch from a causal mask to a bidirectional mask during the weakly supervised fine-tuning stage, the training may initially converge quickly due to the low rank but is prone to getting stuck in local minima, making further optimization challenging.

As illustrated in Figure 2, to address these gaps, we introduce a novel soft mask mechanism. Firstly, to address the issue of attention weights, we introduce an $\alpha(t)$ term in the soft mask, where $\alpha(t)$ is our scheduling function, allowing the mask to gradually transition from 0 to 1, enabling the model to

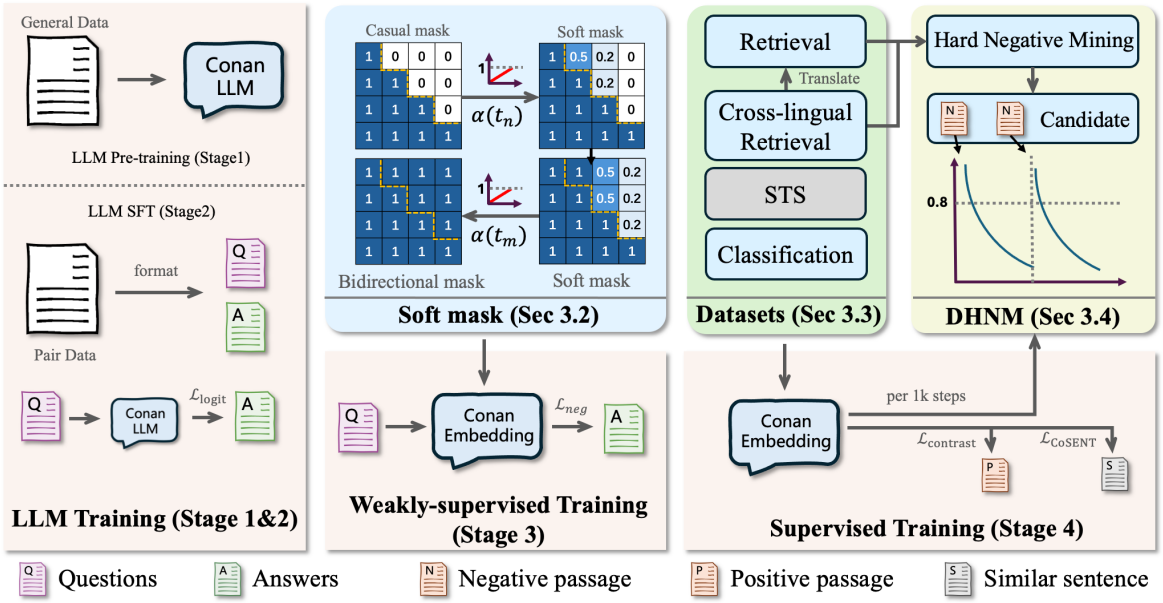


Figure 2: Overview of Conan-embedding-v2: Our approach consists of four stages. During LLM training (stages 1 and 2), we add embedding data to better align the LLM with embedding tasks. In the weakly-supervised training stage, we use the same pairs from LLM SFT and apply a soft mask to bridge the gap between LLMs and embedding models. In the supervised training stage, benefiting from LLM training, we incorporate a cross-lingual retrieval dataset and a dynamic hard negative mining approach to improve data diversity and value.

progressively update these parameters. τ is set to the total number of steps for normalization. $\alpha(t)$ is defined as follows:

$$\alpha(t) = \frac{t}{\tau} \tag{3}$$

Secondly, as weakly supervised training requires learning richer feature representations, we propose a dynamic rank reduction approach. We use M_{ij} to represent the mask matrix. We employed a simple method where the values of the first i column of M_{ij} are set to 1, resulting in a rank of $N - i$. By combining this with our weight adjustment method, the values closer to the beginning transition to 1 more quickly. The formula for the soft mask is as follows:

$$M_{ij}(t) = \begin{cases} 1 & \text{if } i \geq j \\ \min\left(\alpha(t) \times \frac{l}{i}, 1\right) & \text{if } i < j \end{cases} \tag{4}$$

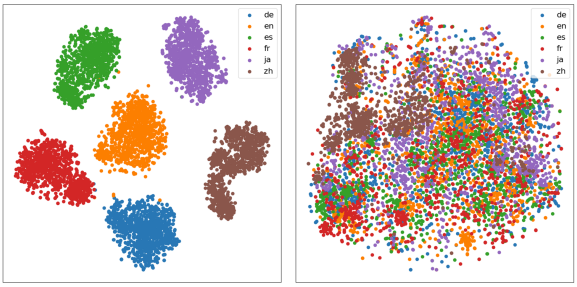
$i < j$ indicates that we are modifying the upper triangular values. l is the training sequence length. We ensure that the maximum value is 1, and the earlier columns reach 1 sooner. This not only allows the rank to gradually decrease but also aligns with the trend of reading from front to back, where the weights gradually decrease. We will discuss the impact of different $\alpha(t)$ on the results in Appendix C.

3.3 Cross-lingual Retrieval Dataset

To develop a multilingual LLM, we aim for Conan-embedding-v2 to learn representations across different languages. Previous work has primarily focused on fine-tuning using multilingual corpora directly or using parallel corpora where the texts are translations, often overlooking the intrinsic relationships between languages. To address this issue, we propose a cross-lingual retrieval dataset (CLR), which integrates representations across different languages through cross-lingual search, thereby narrowing the representation gap between them.

We start with existing retrieval datasets and extend them to support cross-lingual retrieval. To reduce the workload, we only translate the query portion of the datasets using Qwen2.5-7B (Team, 2024). For instance, we translate the queries in MSMARCO (Nguyen et al., 2016) (an English retrieval task) subset to Chinese to enable Chinese-to-English retrieval. Similarly, we apply this approach to other tasks, translating queries to support cross-retrieval among 26 languages, resulting in a total of approximately 10 million pairs.

To provide a more intuitive representation of the embeddings, we conducted a comparative analysis of the embedding distribution. We utilized the Multilingual Amazon Reviews Corpus (Keung et al., 2020), which is not included in our cross-lingual



(a) Vanilla t-SNE Embeddings **(b)** Our t-SNE Embeddings

Figure 3: Embedding distribution before and after training on cross-lingual retrieval dataset.

retrieval dataset. This corpus includes reviews in English, Japanese, German, French, Chinese, and Spanish. For each language, we sampled 1000 sentences from the test set. As shown in Figure 3, the vanilla method represents our model without the CLR dataset included. The embeddings for six different languages were distinctly clustered, each occupying a separate region in the distribution space. In contrast, our model Conan-embedding-v2 successfully integrated the embeddings of all languages into a unified distribution, demonstrating its effectiveness in creating a more cohesive multilingual representation.

3.4 Dynamic Hard Negative Mining

Previous work has primarily focused on hard negative mining during the data preprocessing stage using existing embedding models, resulting in fixed hard negatives. However, the hard negatives identified by other models may differ from those identified by the model being trained. Furthermore, as training progresses and model weights are updated, the score of hard negatives corresponding to the current weights change. Hard negatives mined during the preprocessing stage may become less challenging after several training iterations.

Based on this insight, we propose a dynamic hard negative mining (DHNM) method in conan-v1 (Li et al., 2024b). DHNM dynamically detects the difficulty of the current sample during the training process and replaces the sample based on its difficulty. We use scores to represent the difficulty level, the formula is as follows:

$$S = \cos\langle f(q), f(p)\rangle \tag{5}$$

S represents the cosine score, $f_k(q)$ is query embedding, and $f(p)$ is hard negative embedding.

Unlike the replacement criteria in v1, in this paper, if the absolute value of the score is less than 0.4 at the initial step, it will also be discarded. The current detection formula is:

$$\mathbf{N}_i = \begin{cases} \mathbf{N}_{i+1} & (S_0 < 0.4) \\ \mathbf{N}_{i+1} & 1.2 \cdot S_i < S_0 \ \&\ S_i < 0.7 \\ \mathbf{N}_i & \text{otherwise} \end{cases} \tag{6}$$

N_i denotes the i-th hard negative sample, with $S_{i,0}$ as its initial score and S_i as its current score. If the score multiplied by 1.2 is less than the initial score and the absolute value of the score is less than 0.7, we consider the negative example no longer difficult. We replace it with a new hard negative N_{i+1} from the hard negative pool.

Additionally, in v1, the check is performed every 1k steps. In this paper, we leverage the fact that the similarity score between the query and each hard negative is already computed as part of the loss calculation. During each loss computation, we add a lightweight check to cache the current score of each hard negative and determine whether it is still sufficiently challenging for the model. If a hard negative's score indicates that it is no longer difficult, we mark it for replacement. In the next training step, we replace this negative with a new hard negative sampled from the candidate pool. This process ensures that the set of hard negatives remains up-to-date and challenging throughout training, without introducing additional computational overhead.

4 Experiments

4.1 Training Data

To achieve the multilingual capability of Conan-embedding-v2, we collected large and diverse data for weakly supervised pre-training and embedding fine-tuning. For **weakly supervised pre-training**, we primarily gathered title-content pairs from news articles and websites, specifically from CC-News (Hamborg et al., 2017), mC4 (Karpukhin et al., 2020), Wikipedia and Chinese Corpora Internet (BAAI, 2023). To ensure data integrity, we applied the Data-Juicer (Chen et al., 2024a) tool for systematic removal of low-quality samples, redundant duplicates, and potentially harmful content.

Embedding supervised training. For both Chinese and English, we compiled datasets for five different tasks: retrieval, reranking, classification, clustering semantic textual similarity (STS). We ensured that any training data matching the MTEB evaluation set was filtered out. Detailed data usage is provided in Appendix B.

Languages	Embedding Task Metric	Class. Acc.	Clust V-Meas.	PairClass AP	Rerank MAP	Retri nDCG@10	STS Spear.	Summ. Spear.	Avg.
English	e5-mistral-7b-instruct	79.85	51.44	88.42	49.78	57.62	84.32	36.57	67.97
	stella-en-1.5B-v5	89.38	57.06	88.02	50.19	52.42	83.27	36.91	69.43
	NV-Embed-v2	87.19	47.66	88.69	49.61	62.84	83.82	35.21	69.81
	gte-Qwen2-7B-instruct	88.52	58.97	85.9	**50.47**	58.09	82.69	35.74	70.72
	jasper-en-v1	90.27	**60.52**	88.14	50	56.05	84.37	37.19	71.41
	gemini-embedding-exp-03-07	90.05	59.39	87.70	48.59	64.35	**85.29**	**38.28**	73.30
	Conan-embedding-v2	**90.98**	59.96	**92.35**	49.07	**66.24**	85.12	35.48	**73.52**
Chinese	e5-mistral-7b-instruct	72.96	52.30	66.31	61.38	61.75	48.34	-	59.92
	gte-Qwen2-1.5B-instruct	72.53	54.61	79.50	68.21	71.86	60.05	-	67.12
	bge-multilingual-gemma2	75.31	59.30	79.30	68.28	73.73	55.19	-	67.64
	gte-Qwen2-7B-instruct	75.77	66.06	81.16	69.24	75.70	65.20	-	71.62
	xiaobu-embedding-v2	76.53	65.17	85.94	72.58	76.49	64.18	-	72.36
	Conan-embedding-v1	76.77	66.33	85.68	72.76	76.67	63.67	-	72.50
	retrieve-zh-v1	**76.88**	66.50	85.98	72.86	76.97	63.92	-	72.71
	Conan-embedding-v2	76.47	**68.84**	**92.44**	**74.41**	**78.31**	**65.48**	-	**74.24**

Table 1: Results for MTEB in English and Chinese.

4.2 Model Architecture

As demonstrated in (Kaplan et al., 2020), under a fixed parameter budget, increasing the number of transformer layers beyond seven results in test loss that remains almost constant. Motivated by this observation, we strategically selected eight transformer layers, thereby allocating more parameters to the hidden size and the number of attention heads. This design choice aims to maximize the model's theoretical representational capacity within the given parameter constraints. Consequently, although our model contains only 1.4 billion parameters, it retains the same hidden size as gte-Qwen2-7B-instruct (Li et al., 2023) (3584 dimensions with 28 hidden layers and 28 attention heads). In addition, our model is configured with 32 attention heads and 8 key-value heads using GQA optimization, 8192 intermediate dimensions for the feed-forward network layers, a maximum context window of 32,768 tokens, and a vocabulary size of 150,000 tokens.

4.3 MTEB Results

In this section, we present the experimental results of our method on the MTEB English and MTEB Chinese benchmarks, and compare it with other SOTA methods.

Results for MTEB in English and Chinese. Table 1 provides a detailed comparison of our method's performance on the MTEB English (classic) benchmark and the MTEB Chinese benchmark. The English benchmark additionally includes a summary (summ.) task, which is similar to the STS task. Both tasks measure sentence similarity using Spearman's correlation coefficient. Conan-embedding-v2 achieves SOTA results in both English and Chinese benchmark, excelling in classification (91.11 in English, and 76.8 in Chinese) and reranking (51.49 in English, and 73.69 in Chinese) through multiple training strategies. This is consistent with the results observed in the Multilingual benchmark. Notably, Conan-embedding-v2 performs slightly worse than other models on the STS tasks, which may be due to the lower proportion of STS data compared to other training task.

Results for MTEB in English in zero-shot. To validate the effectiveness and generalization ability of our proposed method, we followed the data selection strategy of e5-mistral-7b-instruct (Wang et al., 2023) and used only a small portion of the MTEB training dataset for zero-shot training. The selected datasets include MSMARCO (Nguyen et al., 2016), NQ (Kwiatkowski et al., 2019), XQuADRetrieval (Rajpurkar et al., 2016a), FEVER (Thorne et al., 2018), HotpotQA (Yang et al., 2018), MIRACLRetrieval (Zhang et al., 2023), and MrTidyRetrieval (Zhang et al., 2021). Table 2 summarizes our model's performance on the MTEB English benchmark in the zero-shot setting. Compared to Linq-Embed-Mistral (7B), our Conan-embedding-v2 (1.4B) achieves a significant improvement. These findings demonstrate the strong zero-shot performance and efficiency of our approach, even with significantly smaller models, validating our innovations in training from scratch and employing novel soft mask techniques to address representational gaps.

Embedding Task Metric	Zero-shot	Class. Acc.	Clust. V-Meas.	PairClass. AP	Rerank. MAP	Retri. nDCG@10	STS Spear.	Summ. Spear.	Avg.
bge-large-en-v1.5	100%	78.34	48.01	87.13	48.26	55.44	82.79	33.13	65.89
multilingual-e5-large-instruct	95%	75.54	49.89	86.24	48.74	53.47	84.72	29.89	65.53
GIST-Embedding-v0	80%	78.16	48.50	86.33	47.52	53.59	83.35	32.32	65.50
UAE-Large-v1	100%	79.08	47.86	87.25	48.35	55.91	84.37	30.13	66.40
mxbai-embed-large-v1	100%	79.10	47.48	87.20	48.05	55.40	84.42	32.63	66.26
GritLM-7B	95%	81.25	50.82	87.29	49.59	54.95	83.03	35.65	67.07
e5-mistral-7b-instruct	95%	79.85	51.44	88.42	49.78	57.62	84.32	36.57	67.97
text-embedding-005	95%	86.03	51.91	87.62	48.84	58.77	**85.18**	35.05	69.60
SFR-Embedding-Mistral	85%	80.47	54.93	88.59	**50.15**	59.33	84.77	36.32	69.31
Linq-Embed-Mistral	95%	83.00	54.07	88.44	49.44	60.14	84.69	**37.26**	69.80
Conan-embedding-v2	95%	**88.35**	**57.34**	**90.97**	47.21	**63.84**	83.77	35.20	**71.43**

Table 2: Zero-shot MTEB results in English.

4.4 MKQA Benchmark

To evaluate cross-lingual retrieval performance, we conducted comprehensive experiments using the Multilingual Knowledge Questions & Answers (MKQA) benchmark proposed by (Longpre et al., 2021). This benchmark provides professionally translated queries and contains 10,000 question-answer pairs from NQ (Kwiatkowski et al., 2019), aligned across 26 typologically diverse languages (260k question-answer pairs in total).

Following previous works (Izacard et al., 2021; Chen et al., 2024b), we conducted retrieval in NQ (Kwiatkowski et al., 2019) for a given question in a specific language and evaluate whether the English passage is present in the retrieved documents. For multilingual models, we computed nDCG@10 and Recall@k (k=20,100) across all 25 target languages to assess both ranking precision and answer coverage. We present the performance details for all languages in the Appendix D.1.

As shown in Table 3, our proposed Conan-embedding-v2 achieves SOTA performance, outperforming all of the baseline models across all metrics. Notably, it achieves significant improvements of +3.6% R@20 and +5.7% nDCG@10 over the strongest baseline (M3-Embedding), demonstrating superior cross-lingual alignment capability.

4.5 Ablation Study

We systematically evaluated the contributions of individual components in our framework through ablation experiments (Table 4). The isolated Cross-lingual Retrieval task objective (Row 2) improves multilingual performance to 62.69% (+1.96% Multi over SM-only) while maintaining stable single-language scores, demonstrating its targeted capability for cross-lingual representation refinement. Using only Dynamic Hard Negative Mining (Row 3) yields the best language-specific results among single components (71.50% Eng/72.09% Zh), confirming its effectiveness in distinguishing fine-grained semantic boundaries through adaptive negative sampling. The combination of SM+CLR (Row 4) produces a significant multilingual performance leap to 64.45% (+3.56% over SM-only), while SM+DHNM (Row 5) achieves peak language-specific scores before full integration. However, both partial combinations reveal an accuracy tradeoff between multilingual and single-language task. Our complete framework with all components (bottom row) resolves this tradeoff by synergistically combining SM's initialization stability, CLR's cross-lingual alignment, and DHNM's discriminative training, achieving SOTA performance across all tasks. These results validate the synergistic effects of Conan-embedding-v2 components in enhancing the model's overall capabilities.

4.6 Analysis

4.6.1 Practical Considerations

In addition to performance, the practical selection of an embedding model is influenced by many other factors. To better demonstrate how our model ensures both efficiency and applicability, we have also highlighted several other important factors. We select model size, embedding dimension, inference time, and support for Matryoshka Representation Learning (MRL) (Kusupati et al., 2022). MRL indicates whether the model supports embeddings of different dimensions. Inference time is measured in minutes and is based on the results obtained using the English queries from the train set of the Multilingual Amazon Reviews Corpus on a single 910B GPU. Additionally, we provide the MTEB English

Model	R@20	R@100	nDCG@10
BM25	28.1	39.9	25.4
mContriever	56.3	67.9	44.9
text-embedding-v3	62.1	69.5	48.1
e5-mistral	62.4	70.1	47.5
M3-Embedding	68.8	75.5	53.2
Conan-embedding-v2	**72.5**	**80.2**	**59.1**

Table 3: Results of cross-lingual retrieval performance on MKQA benchmark.

SM	CLR	DHNM	Multi	Eng	Zh
✔	✗	✗	61.73	70.41	70.99
✗	✔	✗	62.69	70.94	71.41
✗	✗	✔	61.81	71.50	72.09
✔	✔	✗	64.45	72.14	71.79
✔	✗	✔	63.03	72.78	72.44
✔	✔	✔	**65.17**	**73.52**	**74.24**

Table 4: Results of ablation study on MTEB. SM, CLR and DHNM are defined in Sec 3.

Model	Model Size (million)	Embedding Dim.	Infer Time (min.)	MRL	Avg.
gte-large-en-v1.5	335	1024	1.12	✗	65.89
stella-en-1.5B-v5	1543	1536	5.54	✔	69.43
Linq-Embed-Mistral	6782	4096	30.61	✗	69.80
NV-Embed-v2	7851	4096	33.58	✗	69.81
gte-Qwen2-7B-instruct	7613	3584	31.78	✗	70.72
Conan-embedding-v2	1503	3584	5.14	✔	73.52

Table 5: Comparison of practical factors of different embedding models.

benchmark results as a performance reference.

Table 5 shows a comparison between several representative models and our model. Conan-embedding-v2 stands out by maintaining a balanced model size of 1503 million parameters and an embedding dimension of 3584. Despite its compact size, Conan-embedding-v2 achieves an impressive inference time of just 5.14 minutes, making it one of the fastest models evaluated. Additionally, Conan-embedding-v2 supports MRL, a capability shared only with stella-en-1.5B-v5. However, stella-en-1.5B-v5 has a smaller embedding dimension of 1536 and slightly lower performance, with an average score of 71.19. This highlights Conan-embedding-v2's superior efficiency and effectiveness in practical applications.

4.6.2 Training Gap

Token-level LLM training loss and sentence-level contrastive loss have fundamentally different optimization landscapes. Full fine-tuning forces an abrupt transition between these paradigms, causing representation collapse (Luo et al., 2023). In contrast, LoRA updates only a small subset of parameters, providing a smoother optimization path (Zhang et al., 2024). Table 6 compares MTEB-EN results using different methods on Conan-embedding-v2. The results confirm the findings in (Zhang et al., 2024). However, with soft mask applied, higher LoRA ranks consistently yield better results. This demonstrates that soft mask effectively bridges the gap between LLM's generative training and contrastive learning objectives.

Method	w/o SoftMask	w/ SoftMask
LoRA $r = 16$	72.18	72.12
LoRA $r = 32$	72.08	72.23
LoRA $r = 64$	71.83	72.40
Full fine-tuning	71.50	73.52

Table 6: Results on MTEB English with and without SoftMask.

5 Conclusion

In this paper, we propose Conan-embedding-v2, a new LLM trained from scratch and finetuned as a text embedder. We first address the data and training gaps between LLM and embedding models. By leveraging pairs for LLM training, soft mask for embedding weakly-supervised training, cross-lingual retrieval dataset and dynamic hard negative mining for embedding supervised training, Conan-embedding-v2 achieves SOTA while maintaining a reasonable model size and inference speed.

Embedding models are crucial tools that empower fields like recommendation systems, text matching, and entity recognition. We hope to inspire future research in embedding training methods and aim to explore more applications. In the future, we plan to continue updating our model to improve the performance and extend the capabilities in cross-modal retrieval.

Limitations

A Cross lingual Retrieval Data Analysis

To better understand the effectiveness and limitations of the cross-lingual retrieval dataset construction method proposed in Section 3.3, we analyze the potential impact of language distribution within the dataset.

A.1 Proportion of Different Language Pairs. For cross-lingual retrieval, we employ T2Retrieval for Chinese-to-English retrieval and MSMARCO for multilingual retrieval by translating queries into 26 languages. For our translation process, we referenced the language distribution from the MTEB benchmarks to allocate language pairs, resulting in approximately 1 million pairs as shown in Table 7.

Table 7: Language distribution for translation pairs.

Language	Proportion	Language	Proportion
English	25%	Swedish	2%
Chinese	12%	Thai	2%
Spanish	8%	Malay	2%
French	6%	Turkish	2%
Japanese	6%	Vietnamese	2%
German	5%	Dutch	2%
Russian	5%	Polish	2%
Italian	4%	Hindi	2%
Portuguese	4%	Khmer	1%
Arabic	3%	Finnish	1%
Korean	3%	Hebrew	1%
Bengali	2%	Hungarian	1%
Danish	2%	Norwegian	1%

A.2 Performance on Specific Languages. Performance varies significantly across languages depending on the available resources. Table 8 illustrates the performance metrics for languages with different resource levels evaluated on the MKQA dataset. Mid-resource languages, including Spanish, French, Japanese, German, Russian, Italian, and Portuguese, demonstrate better performance compared to low-resource languages. This performance gap likely stems from the disparity in available training data proportions.

Despite being a high-resource language, Chinese shows lower performance. This is likely due to the unique Chinese-English mapping relationship, which conflicts with MKQA's multilingual-to-English evaluation. In the future, we will focus on improving data processing for low-resource languageså and implementing balanced data sampling.

Table 8: Performance by language resource level on MKQA.

Resource	Proportion	Performance
High-resource	37%	70.6
Mid-resource	45%	73.47
Low-resource	18%	72.19

A.3 Potential Biases. The high proportion of English-Chinese data may lead to inflated performance metrics for cognate languages, potentially introducing biases across different language families and linguistic characteristics. We conducted an evaluation of performance across language families. Table 9 shows that Germanic, Slavic, and Romance languages (all Indo-European) exhibit strong performance. Notably, typologically distant languages like Arabic (65.2%) and Korean (67.5%) perform significantly lower, suggesting that linguistic similarity to English, rather than data volume, is the primary factor influencing model effectiveness. This highlights the challenge of achieving consistent performance across diverse language families.

Table 9: Average performance and data share by language family.

Language Family	Avg. Score	Total Share
Chinese	70.4	37%
Germanic	74.6	36%
Romance	73.9	22%
Slavic	75.5	7%
Arabic	65.2	3%
Korean	67.5	3%
Others	67.7	11%

B. Error Analysis

Embedding models often struggle with numerical inconsistencies in semantically similar content. For example, when searching for "3 fairy tales", the model might give low similarity scores to content containing "5 fairy tales", even though the core content is relevant. This happens because embedding models treat numbers as regular tokens without understanding their quantitative relationship. Future improvements could include incorporating retrieval augmented generation to provide external numerical knowledge, and enriching training data with more numerical variations to enhance the model's understanding of quantitative relationships.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. *Joint Conference on Lexical and Computational Semantics*.

BAAI. 2023. Baai-cci: A comprehensive chinese corpus for ai research. https://data.baai.ac.cn/details/BAAI-CCI. Accessed: 2023-10-10.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. 2024. Lora learns less and forgets less. *arXiv preprint arXiv:2405.09673*.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.

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*.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*.

Daoyuan Chen, Yilun Huang, Zhijian Ma, Hesen Chen, Xuchen Pan, Ce Ge, Dawei Gao, Yuexiang Xie, Zhaoyang Liu, Jinyang Gao, Yaliang Li, Bolin Ding, and Jingren Zhou. 2024a. Data-juicer: A one-stop data processing system for large language models. In *International Conference on Management of Data*.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024b. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Xi Chen, Ali Zeynali, Chico Q Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw A Grabowicz, Scott A Hale, David Jurgens, and Mattia Samory. 2022. Semeval-2022 task 8: Multilingual news article similarity.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Quan Do. 2019. Jigsaw unintended bias in toxicity classification.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Gregor Geigle, Nils Reimers, Andreas Rücklé, and Iryna Gurevych. 2021. Tweac: transformer with extendable qa agent classifiers. *arXiv preprint arXiv:2104.07081*.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please : a generic news crawler and extractor. *Ingénierie Des Systèmes D'information*.

Katharina Hämmerl, Jindřich Libovickỳ, and Alexander Fraser. 2024. Understanding cross-lingual alignment–a survey. *arXiv preprint arXiv:2404.06228*.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. Dureader: a chinese machine reading comprehension dataset from real-world applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*.

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*.

Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence Moss. 2020. Ocnli: Original chinese natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

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*.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, WilliamW. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *Cornell University - arXiv*.

Mandar Joshi, Eunsol Choi, DanielS. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *Cornell University - arXiv*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The multilingual amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. 2022. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, page 453–466.

Ken Lang. 1995. *NewsWeeder: Learning to Filter Netnews*, page 331–339.

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

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

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2020. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. *arXiv preprint arXiv:2008.09335*.

Jingyang Li, Maosong Sun, and Xian Zhang. 2006. A comparison and semi-quantitative analysis of words and character-bigrams as features in chinese text categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL - ACL '06*.

Shiyu Li, Yang Tang, Shizhe Chen, and Xi Chen. 2024b. Conan-embedding: General text embedding with more and better negative samples. *arXiv preprint arXiv:2408.15710*.

Yudong Li, Yuqing Zhang, Zhe Zhao, Linlin Shen, Weijie Liu, Weiquan Mao, and Hui Zhang. Csl: A large-scale chinese scientific literature dataset.

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

Robert Litschko, Oliver Kraus, Verena Blaschke, and Barbara Plank. 2024. Cross-dialect information retrieval: Information access in low-resource and high-variance languages. *arXiv preprint arXiv:2412.12806*.

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.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. Mkqa: A linguistically diverse benchmark for multilingual open domain question answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

AndrewL. Maas, RaymondE. Daly, Peter Pham, Dan Huang, AndrewY. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. *Meeting of the Association for Computational Linguistics*.

Maggie, Phil Culliton, and Wei Chen. 2020. Tweet sentiment extraction. `https://kaggle.com/competitions/tweet-sentiment-extraction`. Kaggle.

Julian McAuley and Jure Leskovec. 2013a. Hidden factors and hidden topics. In *Proceedings of the 7th ACM conference on Recommender systems*.

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

Zhongtao Miao, Qiyu Wu, Kaiyan Zhao, Zilong Wu, and Yoshimasa Tsuruoka. 2024. Enhancing cross-lingual sentence embedding for low-resource languages with word alignment. *arXiv preprint arXiv:2404.02490*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, GregS. Corrado, and J.Michael Dean. 2013. Distributed representations of words and phrases and their compositionality. *Cornell University - arXiv*.

Gabriel de Souza P Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. 2024. Nv-retriever: Improving text embedding models with effective hard-negative mining. *arXiv preprint arXiv:2407.15831*.

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

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human-generated machine reading comprehension dataset.

JamesA. 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 reviews. *Empirical Methods in Natural Language Processing*.

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

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016a. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016b. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

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*.

Jianlin Su. 2022. Cosent.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv: Computation and Language*.

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*.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024a. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Yabing Wang, Le Wang, Qiang Zhou, Zhibin Wang, Hao Li, Gang Hua, and Wei Tang. 2024b. Multimodal llm enhanced cross-lingual cross-modal retrieval. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8296–8305.

He sicheng Wang Yuxin, Sun Qingxuan. 2023. M3e: Moka massive mixed embedding model.

Xiaohui Xie, Qian Dong, Bingning Wang, Feiyang Lv, Ting Yao, Weinan Gan, Zhijing Wu, Xiangsheng Li, Haitao Li, Yiqun Liu, and Jin Ma. 2023. T2ranking: A large-scale chinese benchmark for passage ranking.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020. Clue: A chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*.

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*.

Sheng Zhang, Xin Zhang, Hui Wang, Lixiang Guo, and Shanshan Liu. 2018. Multi-scale attentive interaction networks for chinese medical question answer selection. *IEEE Access*, 6:74061–74071.

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

Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. 2021. Mr. tydi: A multi-lingual benchmark for dense retrieval.

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.

A Implementation details

The model is trained with a maximum input length of 32768 tokens. To enhance efficiency, mixed precision training and DeepSpeed ZERO-stage 1 (Rajbhandari et al., 2020) are utilized. For the LLM pre-training stage, we use AdamW (Loshchilov and Hutter, 2017) optimizer and learning rate of 1e-4, with 0.05 warmup ratio and 0.001 weight decay. The batch size is set to 256. The entire pre-training process employs 64 Ascend 910B GPUs and 219 hours. For the LLM finetune stage, we use AdamW (Loshchilov and Hutter, 2017) optimizer and learning rate of 2e-5, with 0.02 warmup ratio and 0.001 weight decay. The batch size is set to 64. The entire pre-training process employs 16 Ascend 910B GPUs and 38 hours. For the embedding weakly-supervised training stage, We used the same optimizer parameters and learning rate as in the pre-training phase. The batch size is set to 64. The entire pre-training process employs 16 Ascend 910B GPUs and 97 hours. For the embedding supervised training stage, the MRL training representation dimensions are configured as 256, 512, 1024, 1536, 2048, 3072, 3584. The batch size is set to 4 for the retrieval task and 32 for the STS task. We sample 7 negatives for each query for retrieval task. We used the same optimizer parameters and learning rate as in the pre-training phase. The entire fine-tuning process employs 16 Ascend 910B GPUs and takes 13 hours.

B Data details

In Section 3.1.1, we have already introduced the datasets used during the LLM training phase. In Section 4.1, we have discussed the types of datasets used during the embedding weakly-supervised training and supervised training phases.

For Retrieval: We utilized datasets such as TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), NQ (Kwiatkowski et al., 2019), MS-MARCO (Nguyen et al., 2016), PubMedQA (Jin et al., 2019), SQuAD (Rajpurkar et al., 2016b), DuReader (He et al., 2018), SimCSE (Gao et al., 2021), FEVER (Thorne et al., 2018).

For Reranking: We used StackOverFlow DupQuestions (Liu et al., 2018) T2Ranking (Xie et al., 2023), CMedQAv2 (Zhang et al., 2018).

For Classification: We used AmazonReviews (McAuley and Leskovec, 2013b), AmazonCounterfactual (O'Neill et al., 2021), Banking77 (Casanueva et al., 2020), Emotion (Saravia et al., 2018), TweetSentimentExtraction (Maggie et al., 2020), MTOPIntent (Li et al., 2020), IMDB (Maas et al., 2011), ToxicConversations (Do, 2019), Tnews, Iflytek (Xu et al., 2020), Multilingualsentiments (McAuley and Leskovec, 2013a).

For Clustering: We employed {Arxiv/Biorxiv /Medrxiv/Reddit/StackExchange/Thunews/CSL}-Clustering-S2S/P2P (Muennighoff et al., 2022; Geigle et al., 2021; Li et al., 2006; Li et al.), TwentyNewsgroups (Lang, 1995).

For STS: We chose STS12 (Agirre et al., 2012), STS22 (Chen et al., 2022), STS-Benchmark (Cer et al., 2017), AFQMC, QBQTC, Cmnli (Xu et al., 2020) and Ocnli (Hu et al., 2020).

For other languages, we leveraged the training data from Mr.Tydi (Zhang et al., 2021) and MIRACL (Zhang et al., 2023).

Table10 and Table11 shows that approximately 1.766 billion pairs were used during the weakly-supervised phase, and approximately 10.6 million pairs were used during the fine-tuning phase. The weakly-supervised training phase leverages a diverse collection of data sources, including News, Knowledge Base, Social Media, Web Page, Academic Paper, Community QA, and Instruction Datasets, as detailed in Table 10. The supervised training phase, as shown in Table 11, focuses on specialized tasks such as Semantic Textual Similarity (STS), Contrastive Learning of Representations (CLR), Retrieval, and Classification.

Table 10: Overview of the data sources used for embedding weakly-supervised training.

Categories	Data Format	Numbers
News	(title, content)	620M
Knowledge Base	(question, answer)	106M
Social Media	(title, content)	690M
Web Page	(input, output)	70M
Academic Paper	(title, content)	50M
Community QA	(question, answer)	30M
Instruction datasets	(prompt, response)	200M

Table 11: Overview of the data used for embedding supervised training.

Tasks	Data Format	Numbers
STS	(sentence, sentence pairs)	1.8M
CLR	(text, pos text, neg text)	3.0M
Retrieval	(text, pos text, neg text)	3.0M
classification	(text, pos label, neg label)	2.8M

Function	**Multi**	**Eng**	**Zh**
Linear	61.73	70.41	70.99
Accelerating	61.50	70.51	70.81
Decelerating	61.43	70.01	70.37

Table 12: Results of different soft mask functions.

C Soft Mask Function

For the $\alpha(t)$ mentioned in Sec 3.2, this section discusses three specific implementation functions: linear decay, quadratic decay (accelerating), and quadratic decay (decelerating). The specific formulas are as follows:

- Linear function: $\alpha(t) = \frac{t}{\tau}$
- quadratic (accelerating): $\alpha(t) = \left(\frac{t}{\tau}\right)^2$
- quadratic (decelerating): $\alpha(t) = 1 - \left(1 - \frac{t}{\tau}\right)^2$

where t represents the current time step, and τ represents the total number of time steps. We conducted comparative experiments on three different functions, exclusively utilizing the soft mask method and not the other two methods. As shown in Table 12, the Linear method yielded the best results, while the Decelerating method showed a decline in performance.

D More Results

D.1 MKQA Results

In this section, we present the results for all languages on the MKQA benchmark. As shown in Table 13, Conan-embedding-v2 outperforms all baselines on average.

D.2 MTEB Results

In this section, we present additional evaluation results on the MTEB English benchmark and MTEB Chinese benchmarks. As shown in Table 14 and Table 15, Conan-embedding-v2 outperforms all baselines on average.

	BM25	mDPR	mContriever	Multilingual-E5-large	e5-mistral-7b-instruct	text-embedding-v3	M3-embedding	Conan-embedding-v2
ar	13.4	33.8	43.8	59.7	47.6	55.1	63.0	65.2
da	36.2	55.7	63.3	71.7	72.3	67.6	72.0	73.1
de	23.3	53.2	60.2	71.2	70.8	67.6	70.4	72.8
es	29.8	55.4	62.3	70.8	71.6	68.0	70.7	73.2
fi	33.2	42.8	58.7	67.7	63.6	65.5	68.9	71.6
fr	30.3	56.5	62.6	69.5	72.7	68.2	70.8	73.5
he	16.1	34.0	50.5	61.4	32.4	46.3	64.6	66.7
hu	26.1	46.1	57.1	68.0	68.3	64.0	67.9	70.2
it	31.5	53.8	62.0	71.2	71.3	67.6	70.3	73.9
ja	14.5	46.3	50.7	63.1	57.6	64.2	67.9	71.8
km	20.7	20.6	18.7	18.3	23.3	25.7	59.5	62.4
ko	18.3	36.8	44.9	58.9	49.4	53.9	63.3	67.5
ms	42.3	53.8	63.7	70.2	71.1	66.1	72.3	78.4
nl	42.5	56.9	63.9	73.0	74.5	68.8	72.3	75.6
no	38.5	55.2	63.0	71.1	70.8	67.0	71.6	76.9
pl	28.7	50.4	60.9	70.5	71.5	66.1	70.4	76.7
pt	31.8	52.5	61.0	66.8	71.6	67.7	70.6	74.8
ru	21.8	49.8	57.9	70.6	68.7	65.1	70.0	74.3
sv	41.1	54.9	62.7	72.0	73.3	67.8	71.5	74.8
th	28.4	40.9	54.4	69.7	57.1	55.2	70.8	75.9
tr	33.5	45.5	59.9	67.3	65.5	64.9	69.6	75.8
vi	33.6	51.3	59.9	68.7	62.3	63.5	70.9	73.0
zh_cn	19.4	50.1	55.9	44.3	61.2	62.7	67.3	70.4
zh_hk	23.9	50.2	55.5	46.4	55.9	61.4	66.7	71.8
zh_tw	22.5	50.6	55.2	45.9	56.5	61.6	65.6	69.7
Avg	28.1	47.9	56.3	63.5	62.4	62.1	68.8	72.4

Table 13: Recall@20 on MKQA dataset for cross-lingual retrieval in all 25 languages.

	Bge-multilingual-gemma2	Gte-Qwen2-7B-instruct	SFR-Embedding-2R	Stella-en1.5B-v5	bge-en-icl	Conan-embedding-v2
ArguAna	77.37	64.27	62.34	65.27	82.76	88.18
ClimateFEVER	39.47	45.88	34.43	46.11	45.35	44.45
CQADupStack	47.94	46.43	46.11	47.75	47.23	52.11
DBPedia	51.37	52.42	51.21	52.28	50.42	56.33
FEVER	90.38	95.11	92.16	94.83	91.96	92.52
FiQA2018	60.04	62.03	61.17	60.48	58.77	62.16
HotpotQA	83.26	73.08	81.36	76.67	84.98	83.36
MSMARCO	45.71	45.92	42.18	45.22	46.72	52.38
NFCorpus	38.11	40.6	41.34	42	40.69	42.09
Natural Questions	71.45	67.73	73.96	71.8	73.85	82.81
QuoraRetrieval	90.04	90.09	89.58	90.03	91.02	90.58
SCIDOCS	26.93	28.91	24.87	26.64	25.25	30.21
SciFact	72.05	79.06	85.91	80.99	78.33	87.60
Touche2020	30.26	30.57	28.18	29.94	29.67	31.09
TREC-COVID	64.27	82.26	87.28	85.98	78.11	93.87
BIOSSES	85.74	81.37	87.6	83.11	86.35	84.78
SICK-R	82.66	79.28	77.01	82.99	83.7	81.91
STS12	77.71	79.55	75.67	80.09	77.73	84.07
STS13	87.45	88.83	82.94	86.09	85.98	86.7
STS14	83.48	85.73	78.43	87.32	82.94	83.18
STS15	87.63	88.54	85.82	89.13	86.54	86.54
STS16	86.49	85.84	87.15	86.54	87.24	87.52
STS17	91.18	88.93	88.9	91.05	91.82	89.09
STS22	69.02	66.88	67.1	68.01	68.08	69.3
STSBenchmark	87.25	83.63	88.23	88.92	86.14	87.01
SummEval	31.2	31.35	31.4	30.75	30.70	30.64
SprintDuplicateQuestions	79.32	97.62	97.61	97.05	95.04	94.99
TwitterSemEval2015	79.64	77.88	80.58	78.54	78.73	80.34
TwitterURLCorpus	86.95	86.59	88.03	87.58	87.19	89.38
AmazonCounterfactual	98.49	98.87	97.88	97.89	95.12	97.12
AmazonPolarity	96.9	97.31	97.1	96.86	97.14	98.91
AmazonReviews	62.56	61.04	59.36	61.28	61.47	66.01
Banking77	92.53	90.2	90.41	90.41	90.34	91.05
Emotion	92.97	79.45	93.37	84.29	93.31	93.68
Imdb	96.66	96.8	96.7	96.8	96.7	96.9
MassiveIntent	82.05	85.7	85.85	85.83	82.26	88.71
MassiveScenario	84.4	89.97	90.61	90.21	83.92	90.1
MTOPDomain	98.61	98.04	98.1	98.2	96.51	95.76
MTOPIntent	95.51	91.88	91.3	92.78	93.56	96.97
ToxicConversations	85.12	91.14	88.75	93.16	92.77	93.08
TweetSentimentExtraction	78.58	79.7	74.84	78.3	80.6	85.03
Arxiv-P2P	54.91	54.46	54.02	55.44	54.42	56.31
Arxiv-S2S	50.28	51.74	48.82	51.44	49.59	57.03
Biorxiv-P2P	52.64	50.09	50.76	50.68	52.32	52.32
Biorxiv-S2S	49.2	46.56	47.67	48.67	44.36	48.39
Medrxiv-P2P	45.81	46.23	46.66	46.8	46.13	46.19
Medrxiv-S2S	44.11	44.18	44.65	44.65	41.36	46.58
Reddit	56.03	73.55	62.92	72.86	71.2	72.32
Reddit-P2P	65.83	74.13	72.74	75.27	72.17	76.15
StackExchange	66.21	79.86	76.48	80.29	81.29	82.13
StackExchange-P2P	45.74	49.4	48.29	49.57	45.53	53.64
TwentyNewsgroups	70.44	53.91	66.42	61.43	68.51	64.17
AskUbuntuDupQuestions	64.59	67.58	66.71	67.33	64.8	67.46
MindSmallRank	31.79	33.36	31.26	33.05	30.6	33.28
SciDocsRR	87.6	89.09	87.29	89.2	86.9	88.94
StackOverflowDupQuestions	54.9	55.06	55.32	55.25	56.32	56.28
MTEB Average (56)	69.88	70.24	70.31	71.19	71.24	73.09

Table 14: MTEB English benchmark.

	e5-mistral -7b-instruct	gte-Qwen2 -7B-instruct	xiaobu-embedding-v2	Conan-embedding-v1	bge-multilingual-gemma2	gte-Qwen2 -1.5B-instruct	Conan-embedding-v2
CmedqaRetrieval	34.23	48.69	47.14	47.61	42.21	46.97	45.32
CovidRetrieval	73.11	81.04	89.40	92.35	77.46	80.79	79.88
DuRetrieval	87.04	87.44	89.44	88.53	90.46	89.40	88.72
EcomRetrieval	45.95	71.15	70.50	70.99	69.30	62.51	68.12
MMarcoRetrieval	74.84	85.16	82.19	82.25	84.70	83.01	83.45
MedicalRetrieval	52.83	65.59	68.19	67.94	62.02	58.65	62.56
T2Retrieval	80.68	87.73	85.01	83.31	86.26	85.47	84.92
VideoRetrieval	45.34	78.84	80.09	80.40	77.40	68.11	76.55
Ocnli	80.21	90.18	92.84	92.54	86.22	90.13	92.74
Cmnli	72.19	87.48	91.87	91.66	86.91	86.67	89.90
AmazonReviews	47.6	53.55	50.07	50.31	54.34	52.95	53.81
MassiveIntent	72.46	81.09	77.45	78.14	78.19	76.25	80.51
MassiveScenario	76.4	85.74	85.3	86.2	82.58	77.26	86.45
IFlyTek	48.65	54.52	51.76	51.94	49.94	44.85	50.32
JDReview	84.69	86.51	89.08	90.32	88.91	85.82	90.09
MultilingualSentiment	74.64	76.88	79.45	78.58	78.91	77.42	80.17
OnlineShopping	92.56	94.30	94.90	95.07	94.59	93.50	94.19
TNews	50.58	52.97	54.64	55.03	50.26	49.95	58.21
Waimai	87.79	89.47	89.34	89.70	89.26	86.63	88.45
CMedQAv1-reranking	76.82	88.20	90.96	91.39	84.62	88.16	91.81
CMedQAv2-reranking	77.59	89.31	90.41	89.72	85.60	88.12	89.45
MMarcoReranking	24.21	31.65	39.91	41.58	35.43	29.14	41.59
T2Reranking	66.90	67.80	69.03	68.36	67.48	67.43	71.91
AFQMC	38.99	72.25	60.96	60.66	47.17	58.42	60.32
ATEC	43.58	62.62	58.81	58.64	50.75	55.65	59.23
BQ	54.67	81.25	75.08	74.51	62.02	73.85	74.63
LCQMC	75.48	73.81	79.82	79.45	75.95	75.39	80.66
PAWSX	16.81	54.06	47.42	46.60	30.57	42.46	45.17
QBQTC	31.80	31.37	45.14	44.58	38.98	35.15	43.98
STSB	84.77	83.88	82.05	81.24	80.87	79.4	81.15
STS22	63.4	65.77	66.96	67.73	68.68	67.4	68.78
CLSClusteringP2P	44.42	47.07	60.42	60.64	54.65	45.21	64.48
CLSClusteringS2S	42.58	45.99	49.54	52.65	63.68	42.50	62.83
ThuNewsClusteringP2P	64.68	86.08	78.76	77.84	64.32	68.24	76.11
ThuNewsClusteringS2S	57.53	85.11	71.96	74.20	54.57	62.50	73.59
MTEB Average (35)	60.89	71.94	72.43	72.62	68.44	67.75	72.83

Table 15: MTEB Chinese benchmark.