PEG: Towards Robust Text Retrieval with Progressive Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Retrieval augmentation has become an effective solution to empower large language models (LLMs) with external and verified knowledge sources from the database, which overcomes the limitations and hallucinations of LLMs in handling up-to-date and domain-specific information. However, existing embedding models for text retrieval usually have three non-negligible limitations. First, the number and diversity of samples in a batch are too restricted to supervise the modeling of textual nuances at scale. Second, the high proportional noise is detrimental to the semantic correctness and consistency of embeddings. Third, the equal treatment of easy and difficult samples would cause sub-optimum convergence of embeddings with poorer generalization. In this paper, we propose the **P**rogressively learned textual **E**mbeddin**G** (PEG) for robust text retrieval. Specifically, we increase the number of negative samples per training batch to 80,000, with each query paired with at least five hard negatives via offline mining. Concurrently, we incorporate a progressive learning mechanism to enable the model to dynamically modulate its attention to the samples throughout training. Additionally, PEG is trained on more than 100 million data, encompassing a wide range of domains (*e.g.*, finance, medicine, and tourism) and covering various tasks (*e.g.*, question-answering, machine reading comprehension, and similarity matching). Extensive experiments on C-MTEB and DuReader demonstrate that PEG surpasses state-of-the-art embedding models in retrieving true positives, highlighting its significant potential for applications in LLMs. Code and dataset will be released upon acceptance.

1 Introduction

Information (knowledge) retrieval, a crucial aspect of natural language processing, plays an increasing role in the context of large language models (LLMs) [32, 29, 31, 44, 5, 46, 56, 41]. The employment of a retrieval model to incorporate external knowledge is essential to enhancing the accuracy and validity of answers generated by LLMs. Most existing approaches utilize the pipeline of dense passage retrieval (DPR) [9, 34, 18, 10, 36, 28, 19], which include two steps: text encoding and text matching. The encoder of any off-the-shelf language model is used to map queries and a pool of document fragments into representations in the embedding space, and then the similarity between queries and document fragments is measured to match the most relevant candidates.

In the field of text encoding, contrastive learning (CL) has emerged as one of the most intuitively effective methods for training embeddings [10, 18, 30, 48]. This approach aims to minimize the distance between similar, positive sample pairs, while simultaneously maximizing the distance between dissimilar, negative pairs. Given the high cost associated with collecting large-scale labeled corpora, the training process is typically divided into two stages: 1) task-agnostic unsupervised pre-training, and 2) task-specific supervised fine-tuning. During the first stage, methods such as

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

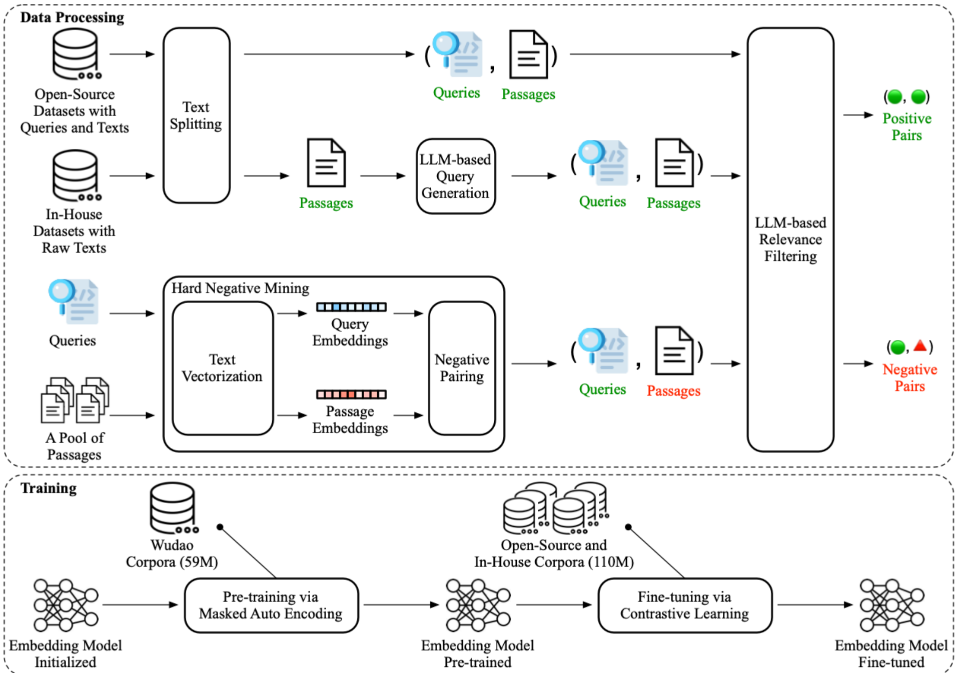

Figure 1: The pipeline of PEG. During data processing, we first split raw texts into shorter passages to control the length of text segments from different datasets. Since our in-house datasets only contain passages without structured queries, we perform query generation based on each passage via an LLM. Then, for each query-passage pair from both open-source and in-house datasets, we perform hard negative mining via a retrieval model to find the five most similar passages as hard negatives. After that, the LLM-based relevance filtering is conducted on positive/negative query-passage pairs to respectively filter out irrelevant/relevant pairs. During training, we first pre-train the embedding model via masked auto-encoding on raw texts of the Wudao Corpora. Then, both positive and negative sample pairs from the open-source and in-house corpora are involved in fine-tuning the embedding model via contrastive learning.

SimCSE [10] employ random augmentation (*e.g.*, dropout) on the output layer to generate two highly similar yet non-identical counterparts. CL is then performed on these two equivalents as a positive pair, while the remaining samples in a batch are paired with the current sample as negatives. In the second stage, human annotations verify positive and negative pairs. Typically, each query is positively associated with only one passage, while all other passages in a batch are considered negatives.

One challenge associated with CL-based embedding learning is that the representation capacity is closely related to the quality and quantity of negative samples. A small batch of negative samples that are not sufficiently high-quality and diverse may fail to effectively compel the model to discern the subtle differences among highly similar samples, thus impeding its ability to achieve superior discrimination. Consequently, BGE [48] substantially increases the batch size by allowing for more than 60,000 negative samples in each batch during training. Moreover, it incorporates a hard negative mining approach for offline data processing, in which numerous negative samples are chosen using external retrieval models. Although BGE tackles the issue of quantity and diversity of negatives, it still possesses limitations as follows: First, BGE does not properly handle the risk of introducing more false negatives while blindly increasing the batch size. A large proportion of noise in a single batch inevitably degrades the effectiveness of embeddings if no intervention is performed to combat noise. Secondly, it assigns equal weight to all negatives and disregards the varying difficulties of learning easy and hard negatives. The overfitting of a majority of simple negatives ultimately leads to sub-optimal convergence.

To further improve the generalization and robustness of the text retrieval model, we introduce the *PEG*, a **P**rogressively learned textual **E**mbeddin**G**. First and foremost, we have amassed an extensive collection of over 110 million data, spanning a wide range of fields such as general knowledge, finance, tourism, and medicine. Our dataset encompasses a diverse array of downstream tasks, including the question-answering (QA) tailored for short text retrieval and the machine reading comprehension (MRC) for long text retrieval. Secondly, for each query, we carefully extract five hard negatives from the dataset to improve the contrastive efficiency. We initially perform an off-line retrieval to obtain the five most similar negatives and then employ an LLM for further cleansing and refinement. If the LLM considers one negative highly similar to the query, this negative sample is filtered out to avoid false negatives. Furthermore, by leveraging massive computational resources, we are capable of accommodating up to 84,000 negative samples within a single batch. Accordingly, we assign varying weights to different negative samples for the progress of training via the measurement of learning difficulty, thereby facilitating the learning procedure. Extensive experiments on various downstream benchmarks showcase the effectiveness of PEG with the state-of-the-art (SOTA) performance.

Our contributions are summarized as follows: **(i)** We collect a large-scale retrieval dataset consisting of 110 million queries, where each query is paired with one positive sample and five carefully selected hard negatives. **(ii)** We propose PEG, which progressively adjusts the weights of samples based on the difficulties of negative samples. **(iii)** Extensive experiments demonstrate that PEG achieves the SOTA performance.

2 Related Work

Dense Text Retrieval The key difference between dense and sparse text retrieval methods lies in the implementation of the retriever model. For the sparse retrieval like BM25 [38, 37], lexical matching is performed while for the dense retrieval like Condenser [9], semantic similarity is measured for matching. Specifically for dense text retrieval, queries and passages are respectively represented as dense vectors. The relevance score between the query and the passage is calculated by similarity measurement between these vectors [52, 8, 11, 24]. REALM [12] developed a latent knowledge retrieval to allow the language model to retrieve and attend over passages in the pre-training and fine-tuning stages. DRPQ [43] generated multiple pseudo query embeddings for the representation of documents and boosted the retrieval performance via the nearest neighbour search. TASER [4] improved the dual-encoder retrieval model via parameter sharing and proposed to interleave the shared and specialized blocks in one encoder.

Contrastive Learning Contrastive learning methods can be roughly categorized as: 1) context-instance contrast, where the relationship of local parts with respect to global context is learned [20]; 2) instance-wise contrast, where similar image pairs are pulled closer with dissimilar pairs pushed farther [13, 3]. PCL [21] encourages each image embedding to be adjacent to its assigned cluster prototype for unsupervised contrastive representation learning. SentenceBERT [35] explicitly learns a sentence embedding using the triplet loss where two sentences from the same passage are considered positive pairs and are negative otherwise.

3 Methodology

In this section, we provide a detailed explanation of the proposed PEG. We begin by introducing the data collection and processing procedure, then followed by a discussion on the pre-training and fine-tuning steps (see Fig. 1).

3.1 Dataset Source

Pre-training. We make use of the publicly available Wudao Corpora [53] for language model pre-training. It is a huge (nearly 59 million) and high-quality Chinese dataset which contains both the title and body of passages.

Fine-tuning. We collected 110 million data for fine-tuning (see Fig. 2). The vast majority of our data comes from open-source datasets while only a small portion of our datasets are privately constructed. The open-source datasets cover a variety of tasks such as text summarization, question answering (QA), and text matching. For the summarization task, we utilize title-passage datasets like Wudao [53], LCSTS [14], WeiXin Public Corpus[1], CSL [23], NLPCC

[1] https://github.com/nonamestreet/weixin_public_corpus

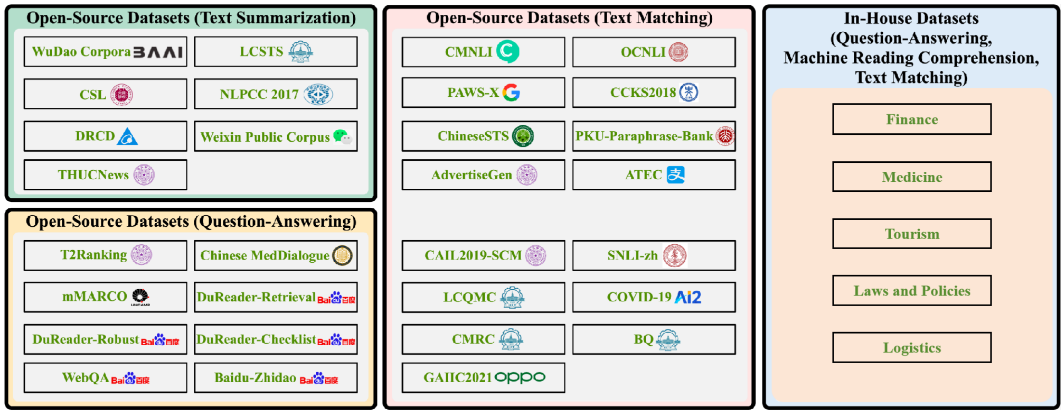

Figure 2: Overview of the datasets. Both open-source and in-house datasets contain a variety of tasks and domains.

2017 [16], DRCD [39], and THUCNews [40]. For the QA task, we utilize datasets like DuReader-Retrieval [33], WebQA [22], T2Ranking [49], mMARCO [1], Chinese Medical Dialogue [2], BaiDu-Zhidao[2], DuReader-Robust [42], and DuReader-Checklist[3]. For the text matching task, we use CMNLI [50], OCNLI [15], LCQMC [25], PAWS-X [51], CCKS2018 [55], COVID19 [45], ChineseSTS[4], CMRC [6], AdvertiseGen[5], ATEC[6], BQ[7], GAIIC2021-OPPO[8], CAIL2019-SCM [47], and PKU-Paraphrase-Bank [54]. Our in-house datasets are primarily composed of high-quality books and journals, covering domains of finance, medicine, tourism, laws and policies, and logistics.

3.2 Data Processing

Pre-training. We follow BGE [48] to directly use the raw texts in Wudao Corpora for pre-training without additional pre-processing.

Fine-tuning. To control the length of articles and chapters from various corpora for efficient and effective retrieval, we first perform text splitting on raw texts to obtain shorter passages. In contrast to most open-source datasets where the structured pairs of queries and passages are available, our curated in-house datasets only contain plain texts without manual annotation. Therefore, we make full use of an off-the-shelf LLM (*e.g.*, GPT4 [31]) to generate questions as queries for each short passage. Each query-passage pair is treated as a positive sample pair. Meanwhile, we perform the off-line hard negative mining to allocate five hard negatives to each query. Specifically, we utilize an open-source retrieval model (*e.g.*, Text2Vec[9]) to pinpoint the five most similar passages (excluding the paired positive one) for each query. These passages are paired with the query as negative sample pairs. After that, we further employ an LLM to clean up all the available sample pairs via relevance filtering. Given each query-passage pair, the LLM determines if the query is positively or negatively associated with the passage. If a positive sample pair is considered irrelevant, we directly discard all query-passage pairs containing the same query in this pair. If a negative sample pair is deemed relevant, we perform sampling with replacement on the remaining negative pairs associated with the same query. This is to guarantee that the total count of hard negative pairs equals five for each specific query.

[2]https://github.com/liuhuanyong/MiningZhiDaoQACorpus
[3]https://github.com/baidu/DuReader/tree/master/DuReader-Checklist
[4]https://github.com/IAdmireu/ChineseSTS
[5]https://huggingface.co/datasets/shibing624/AdvertiseGen
[6]https://github.com/IceFlameWorm/NLP_Datasets/tree/master
[7]http://icrc.hitsz.edu.cn/info/1037/1162.htm
[8]https://tianchi.aliyun.com/competition/entrance/531851/introduction
[9]https://huggingface.co/shibing624/text2vec-base-chinese

3.3 Training

Pre-training. Our model is pre-trained on the Wudao corpora. We utilize the MAE-style approach, as presented in RetroMAE [26], to train the model effectively. The corrupted text $\hat{X}$ is transformed into its embedding representation, from which the clean text X is reconstructed using a lightweight decoder. The objective of pre-training can be defined as follows:

$$\mathcal{L}_{pt} = \sum_{x \in X} -\log \mathrm{Dec}(x | \boldsymbol{e}_{\hat{X}}), \boldsymbol{e}_{\hat{X}} \leftarrow \mathrm{Enc}(\hat{X}), \tag{1}$$

where the Enc and Dec are respectively abbreviations of the encoder and decoder respectively.

Fine-tuning. The pre-trained model is fine-tuned using contrastive learning, which improves the model's capacity to differentiate text pairs by minimizing the distance between positive sample pairs and maximizing the separation between negative pairs. We employ the widely-used InfoNCE loss[13] for model optimization:

$$\begin{aligned} \mathcal{L}_{ft} = \sum_{(\boldsymbol{e}_q, \boldsymbol{e}_p)} -\log \frac{h(\boldsymbol{e}_q, \boldsymbol{e}_p)}{h(\boldsymbol{e}_q, \boldsymbol{e}_p) + \sum_n^N h(\boldsymbol{e}_q, \boldsymbol{e}_n)}, \\ h(\boldsymbol{e}_q, \boldsymbol{e}_p) = \exp(\mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_p)/\tau), \\ h(\boldsymbol{e}_q, \boldsymbol{e}_n) = \exp(\mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_n)/\tau), \end{aligned} \tag{2}$$

where q and p represent the indices of a query and its corresponding positive sample, respectively. The index of a negative sample is $n \in \{1, 2, ..., N\}$, where N denotes the total number of negative samples. Accordingly, the embeddings $(\boldsymbol{e}_q, \boldsymbol{e}_p)$ are positive sample pairs, while $(\boldsymbol{e}_q, \boldsymbol{e}_n)$ are negative ones. τ is the temperature hyper-parameter. We use $\mathrm{s}(\cdot)$ to represent the similarity measurement (*e.g.*, cosine similarity) between sample pairs.

One non-negligible disadvantage of the InfoNCE loss above is that it overlooks the difficulty of learning various positive and negative samples. Negative samples exhibit diverse patterns and the degree of their resemblance to the query indicates how difficult it is for the model to learn to identify their distinction. The overfitting of the dominating easy negatives would weaken the validity of the contrast.

Under such circumstances, each negative pair ought to make an unique contribution to the polishing of embeddings. We consequently propose the progressive learning mechanism to assign adaptive weights to sample pairs of different levels of learning difficulty. It enables the embedding model to focus on simple samples in the initial stages to first gain preliminary knowledge. Then, it gradually shifts the model's attention towards more challenging samples as the training progresses. Given one mini-batch of B positive pairs and N negative pairs, our objective is defined as follows:

$$\begin{aligned} \mathcal{L}_{ft} = \sum_{(\boldsymbol{e}_q, \boldsymbol{e}_p)} -w_q \log \frac{h(\boldsymbol{e}_q, \boldsymbol{e}_p)}{h(\boldsymbol{e}_q, \boldsymbol{e}_p) + \sum_n^N g(a_n, \boldsymbol{e}_q, \boldsymbol{e}_n)} \\ g(a_n, \boldsymbol{e}_q, \boldsymbol{e}_n) = \exp(a_n \cdot \mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_n)/\tau), \end{aligned} \tag{3}$$

where the weight w_q and the scaling factor a_n are defined below respectively:

$$\begin{aligned} w_q &= \begin{cases} 1, & \text{if } \mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_p) \geq \sigma, \\ \mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_p)/\sigma & \text{otherwise}, \end{cases} \\ \sigma &= \frac{1}{B} \sum_{(\boldsymbol{e}_q, \boldsymbol{e}_p)} \mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_p) - \beta, \end{aligned} \tag{4}$$

$$a_n = \begin{cases} 1, & \text{if } \mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_p) < \sigma \text{ or} \\ & \mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_n) < \mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_p), \\ t + \mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_p), & \text{otherwise}, \end{cases} \tag{5}$$

where σ is a threshold and β is its margin. We measure the similarities of all positive sample pairs within a batch as the normalization basis of the current positive pair. The hyper-parameter t acts as a bias with respect to the similarity between e_q and e_p. Compared with the vanilla InfoNCE loss (Eq. 2), we intuitively consider that the positive sample pairs whose similarity is below a threshold are potentially false positives and therefore their contribution to the total loss should be weighted according to the batch-wise statistics (*e.g.*, the averaged similarity). Besides, we calibrate the dissimilarities between negative pairs for loss penalty by comparing the similarity between each negative pair and the positive pair. If one negative sample highly resembles the query, it is reasonable to believe that such a negative is a hard one and consequently extra emphasis should be put on learning the nuances between the query and this negative.

When it comes to the proper scaling for such calibration, one naive solution is to set a constant as the bias term t. However, motivated by the momentum mechanism [13, 17], we further bring in the batch-wise statistics with a consistent and smooth update policy:

$$t^{(s)} = \alpha \cdot \frac{1}{B} \sum_{(\boldsymbol{e}_q, \boldsymbol{e}_p)} \mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_p) + (1-\alpha) \cdot t^{(s-1)}, \tag{6}$$

where $t^{(s)}$ refers to the update of t at the s-th step during training and α denotes the momentum coefficient. Initially, we set $t^{(0)}$ to 0.

With the progress of training, the scaling factor would not only reflect the overall similarity distributions across batches but also retain the description of the current positive pair. Given the proposed progressive learning mechanism, the optimization of embeddings can greatly benefit from the large-scale contrastive learning to improve their discriminability and robustness against noise.

4 Experiments

We conducted experiments on two Chinese text retrieval benchmarks and one Chinese text reranking benchmark.

Table 1: Results on the retrieval task of C-MTEB are reported in NDCG@10.

Model	T2 Retrieval	MMarco Retrieval	Du Retrieval	Covid Retrieval	Cmedqa Retrieval	Ecom Retrieval	Medical Retrieval	Video Retrieval	Avg
Text2Vec (base)	51.67	44.06	52.23	44.81	15.91	34.6	27.56	39.52	38.80
Text2Vec (large)	50.52	45.96	51.87	60.48	15.53	37.58	30.93	42.65	41.94
Text2Vec-bge (large)	48.64	30.06	51.36	41.22	22.27	31.08	33.08	41.38	37.38
M3E (base)	73.14	65.46	75.76	66.42	30.33	50.27	42.79	51.11	56.91
M3E (large)	72.36	61.06	74.69	61.33	30.73	45.18	48.66	44.02	54.75
SimCSE	27.98	32.52	36.58	34.06	13.71	14.07	22.07	20.4	25.17
Contriever	33.55	44.37	38.24	37.34	14.53	35.67	23.44	41.3	33.56
OpenAI-Ada-002	69.14	69.86	71.17	57.21	22.36	44.49	37.92	43.85	52
BGE (base)	83.35	79.11	86.02	72.07	41.77	63.53	56.64	73.76	69.53
BGE (large)	84.82	**81.28**	**86.94**	74.06	42.4	66.12	59.39	**77.19**	71.53
PEG	**84.94**	81.04	86.84	**82.14**	**42.57**	**66.4**	**60.66**	76.53	**72.64**

4.1 Datasets

C-MTEB. The Chinese Massive Text Embedding Benchmark (C-MTEB) [48] is presently the most comprehensive evaluation benchmark for Chinese semantic embeddings. It encompasses 6 evaluation tasks, namely the retrieval, reranking, sentence similarity, reasoning, classification, and clustering. We mainly focus on the retrieval and reranking. It is noted that the reranking task can also be viewed as another kind of retrieval as it retrieves the true positives from a pool of candidates that share high similarity with respect to the query.

The retrieval task predominantly encompasses the following datasets: T2Retrieval, MMarcoRetrieval, DuRetrieval, CovidRetrieval, CmedqaRetrieval, EcomRetrieval, MedicalRetrieval, and VideoRetrieval. Both the EcomRetrieval and VideoRetrieval pertain to sentence-level keyword matching and retrieval, whereas the rest focus exclusively on query-to-passage retrieval. For the reranking task, we use T2Reranking, MmarcoReranking, CMedQAv1, and CMedQAv2.

DuReader-Retrieval. The DuReader-Retrieval dataset [33] contains a training set, development set, and test set with the original paragraph corpus. It is the first large-scale high-quality Chinese paragraph retrieval dataset based on user search logs under real scenarios. The queries in the dataset are all real user questions from the Baidu search engine, and the passages in the dataset are all collected from the retrieved results of Baidu. We evaluate the performance of our model on the development set, which contains 2,000 query samples and a total of 8.09 million paragraphs.

4.2 Evaluation Metrics

For the C-MTEB retrieval task, we employ the normalized discounted cumulative gain (NDCG)@10 as our evaluation metric, with the primary objective of concentrating on the accuracy of ranking within the top 10 recall results. For the C-MTEB reranking task, the mean average precision (MAP) score is used as the main metric. And for the DuReader-Retrieval, both the mean reciprocal rank (MRR) and Top-K recall (Recall@K) are adopted. Specifically, we use the MRR@10 of the top 10 retrieved passages, the recall rate of the top 1 retrieved passages (Recall@1), and the recall rate of the top 50 retrieved passages (Recall@50).

Table 2: Results on the reranking task of C-MTEB are reported in the mean average precision (mAP).

Model	T2 Reranking	Mmarco Reranking	CMedQA v1	CMedQA v2	Avg
Text2Vec (base)	65.95	12.76	59.26	59.82	49.45
Text2Vec (large)	64.82	12.48	58.92	60.41	49.16
Text2Vec-bge (large)	63.51	9.24	63.42	63.57	49.94
M3E (base)	66.13	16.46	77.76	78.27	59.66
M3E (large)	66.03	17.51	77.05	76.76	59.34
SimCSE	61.34	12.38	57.04	57.72	47.12
Contriever	62.16	13.57	49.82	52.28	44.46
OpenAI-Ada-002	66.65	23.39	63.08	64.02	54.28
BGE (base)	66.49	28.24	80.11	84.78	64.91
BGE (large)	66.2	26.23	83.01	85.01	65.11
PEG	**68.89**	**32.03**	**84.08**	**85.14**	**67.53**

4.3 Implementation details

We use the BERT-large [7] model as our basic model architecture. We train our model on 32 H800 GPUs. For pre-training, we use AdamW [27] optimizer, with an initial learning rate of 2e-5 and a linear decay applied to the learning rate. The batch size per GPU is set at 32, the maximum input sequence length is 512, and the model is trained for 3 epochs. For the fine-tuning phase, we employ the same optimizer and learning rate decay pre-training stage. The initial learning rate is set to 1e-5, with a batch size of 432 per GPU. The maximum sequence lengths for the input query and document are 64 and 256 respectively. And the model is trained for 5 epochs. In addition, we refer to BGE to add an instruction in front of each query sample for better retrieval performance. Out of simplicity, we empirically set $\alpha = 0.5$, $\beta = 0.1$, and $\tau = 0.01$.

4.4 Experimental Results

C-MTEB. The results on C-MTEB retrieval task and reranking task are shown in Table 1 and 2 respectively. In the retrieval task, the newly proposed PEG model attains the SOTA performance, as evidenced by the average NDCG@10 across eight distinct datasets. Notably, the PEG method surpasses existing methods by a large margin on the CovidRetrieval dataset. As a query-to-passage retrieval dataset, CovidRetrieval consists of passages extracted from comprehensive articles rather than short answers to queries. The key information pertinent to the query within extensive texts tends to be more dispersed, thereby increasing the complexity of this dataset. This necessitates the use of high-performing embeddings capable of accurately capturing fine-grained semantics. In the context of the reranking task, PEG continues to demonstrate the SOTA results across all evaluated datasets.

DuReader-Retrieval. The DuReader-Retrieval development set contains 2,000 queries that require our model to pinpoint the most relevant passage from the extensive gallery corpus of over 8 million documents. To conserve computational resources, we have randomly selected a subset of 200,000 documents from the original gallery to create a new smaller gallery. As shown in Table 3, our PEG significantly exceeds other models in all metrics. Compared with the BGE (large) model, our model achieves an increase of around 4% in Recall@1 and 2% in MRR@10. It's worth noting that the size

Table 3: Results on the evaluation set of Du-Retrieval.

Model	Dureader-Retrieval (200,000 documents)		
	MRR@10	Recall@1	Recall@50
Text2Vec (base)	56.29	44.70	89.45
Text2Vec (large)	60.28	49.35	89.75
Text2Vec-bge (large)	61.88	52.60	87.10
M3E (base)	75.36	65.3	96.05
M3E (large)	76.95	67.5	96.65
SimCSE	48.26	38.35	79.8
Contriever	50.74	39.55	85.6
BGE (base)	85.39	77.85	97.80
BGE (large)	87.09	80.25	98.45
PEG	**89.27**	**84.10**	**98.50**

of DuReader-Retrieval's gallery corpus is 200,000, which doubles the average size of the C-MTEB (100,000). Despite such a more challenging gallery, our model still outperforms the SOTA methods. Without losing generalization, we also compared our model with BGE (large) on the performance of retrieval with the full gallery of 8 million documents in Table 4. We observe a clear advantage over BGE. Specifically, we achieved 6.45% and 7.65% improvements in MRR@10 and Recall@1, respectively. This further illustrates that under a more complex and strict condition, our model is relatively more robust and consistent.

Table 4: Results on the evaluation set of Du-Retrieval.

Model	Dureader-Retrieval (8 million documents)		
	MRR@10	Recall@1	Recall@50
BGE (large)	44.89	32.2	**92.6**
PEG	**51.34**	**39.85**	91.65

4.5 Ablation Study

In this section, we carry out a series of experiments to assess the efficacy of PEG on C-MTEB. To conserve computational resources and enhance efficiency, we randomly selected a sample of 10 million (10M) data points from the original dataset for our ablation studies.

Effectiveness of Data Cleaning To validate the efficacy of our data cleansing procedure, we initially utilize the standard InfoNCE, in accordance with Formula 2, to set up a baseline, as depicted in the first row of Table 5. Subsequently, we employ a sophisticated language model to evaluate the correlation between each pair. In the end, we were able to refine and retain 8.9M of pristine data from the original 10M dataset. By leveraging only 8.9M of this more refined data, the model's performance notably improved from 64.34% to 65.26%. From the experimental results, it can be observed that the performance with 8.9M data volume even surpasses that of 10M, which attests to the effectiveness of the data cleaning process.

Table 5: Effectiveness of PEG on C-MTEB retrieval.

Model	C-MTEB Retrieval
Effectiveness of data cleaning	
baseline	64.34
+ Data correlation cleaning	65.26 (+0.92)
Effectiveness of progressive learning	
PEG	**66.33** (+1.99)
- loss weight w_q	66.10 (-0.23)
- scale factor a_n	65.81 (-0.52)

Effectiveness of Progressive Learning We then utilize the obtained 8.9M data to evaluate the effectiveness of each hyperparameter within the progressive learning mechanism. Notably, with the use of the 8.9M data, PEG achieves a performance of 66.33%. However, in the absence of w_q which controls the weight of the loss based on the similarity of the positive pairs, we observed a performance decrease of 0.23%. Furthermore, if we eliminate the key scale factor a_n that governs the weight of negative samples, the performance experiences a more significant drop of 0.52%. These findings substantiate that PEG effectively weights the loss and hard samples based on the difficulty of positive and negative sample pairs, thereby enhancing the effectiveness of progressive learning.

To verify the critical role of the progressive learning mechanism in enhancing model retrieval performance, we conducted further ablation studies on the English benchmark. The experimental

Table 6: Effectiveness of Progressive Learning on the English Benchmark. (NDCG@10)

Method	TREC-COVID	SCIDOCS	ARGUANA	HOTPOTQA	NFCORPUS	NQ	FIQA	SCIFACT	FEVER
PEG	27.12	12.96	28.99	70.35	21.67	26.59	21.47	41.06	71.65
Baseline	26.17	11.84	27.72	67.52	21.21	26.45	20.01	37.82	68.32

results are shown in Table 6. The baseline model and our method used the same training data and experimental settings to fairly demonstrate the impact of progressive learning on performance. On the TREC-COVID, SCIDOCS, ARGUANA, and FIQA benchmarks, the model employing progressive learning showed nearly a 1-point improvement in the NDCG@10 metric. Additionally, on the HOTPOTQA, SCIFACT, and FEVER benchmarks, the performance of the model using the progressive learning framework significantly surpassed that of the baseline model. This not only demonstrates the effectiveness of the PEG method in multilingual scenarios but also highlights that the progressive learning approach can significantly improve retrieval performance even when the model is trained with standard data.

The impact of hyperparameter β When the similarity of a positive pair dips below a certain threshold, we interpret it as noise or a false positive. We then assign it a relatively lower weight to lessen its overall influence. The hyperparameter β is employed to adjust the margin of this threshold. As shown in Figure 3, when β equals 0, the threshold is relatively high, causing more positive samples to be classified as noise, which subsequently leads to a drop in performance. Conversely, when β is set to a relatively high value of 0.5, nearly all positive samples surpass the threshold, rendering w_q ineffective. However, when β is adjusted to an optimal value, specifically 0.2, we achieve the best results.

Change Trend of Scale Factor We utilize a scale factor to re-adjust the weights of challenging negative samples. As depicted in Figure 4, θ_n represents the angle between the negative sample and its corresponding query. Within a certain similarity range, such as $3\pi/16$ to $\pi/2$, the weight of the negative sample exhibits a positive correlation with its degree. However, when the similarity surpasses this interval, the sample is considered more difficult or a false negative, and the weight begins to show a negative correlation. During the initial stages of training, the model is more focused on simpler samples (with bias term t=0), hence the weight of difficult samples is relatively smaller. This weight gradually increases as the training process advances (with bias term t=0.58).

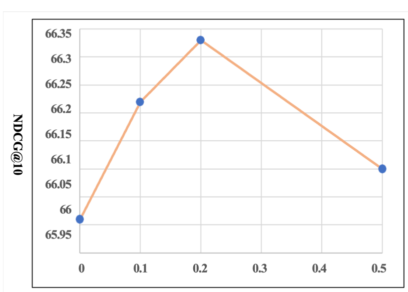

Figure 3: The impact of hyperparameter β

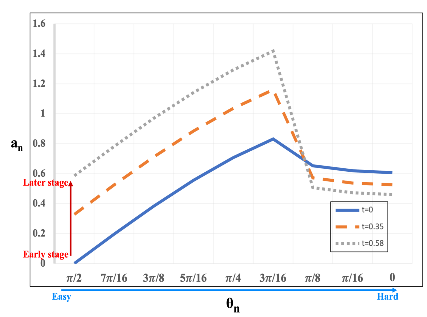

Figure 4: The trend of negative sample weights

5 Conclusion and Limitations

In this paper, we propose PEG for robust text retrieval. Addressing the limited number and diversity of samples, especially the negatives, we prepare a large-scale dataset across a variety of domains and tasks. We increase the batch size up to 80,000 to enable effective contrastive learning. In addition, we've dedicated particular attention to hard negative mining, incorporating a curriculum strategy that progressively assigns adaptive weights to samples based on their level of difficulty at various stages of training. Comprehensive experiments validate that PEG attains SOTA performance in text retrieval and reranking tasks. In the future, we will carry out evaluations on datasets encompassing a broader range of languages.

References

[1] Luiz Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. mmarco: A multilingual version of the ms marco passage ranking dataset. *arXiv preprint arXiv:2108.13897*, 2021. 4

[2] Shu Chen, Zeqian Ju, Xiangyu Dong, Hongchao Fang, Sicheng Wang, Yue Yang, Jiaqi Zeng, Ruisi Zhang, Ruoyu Zhang, Meng Zhou, et al. Meddialog: a large-scale medical dialogue dataset. *arXiv preprint arXiv:2004.03329*, 3, 2020. 4

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020. 3

[4] Hao Cheng, Hao Fang, Xiaodong Liu, and Jianfeng Gao. Task-aware specialization for efficient and robust dense retrieval for open-domain question answering. *arXiv preprint arXiv:2210.05156*, 2022. 3

[5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 1

[6] Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. A span-extraction dataset for chinese machine reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5883–5889, 2019. 4

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 7

[8] Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, et al. Pre-training methods in information retrieval. *Foundations and Trends® in Information Retrieval*, 16(3):178–317, 2022. 3

[9] Luyu Gao and Jamie Callan. Condenser: a pre-training architecture for dense retrieval. *arXiv preprint arXiv:2104.08253*, 2021. 1, 3

[10] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021. 1, 2

[11] Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–42, 2022. 3

[12] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR, 2020. 3

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 3, 5, 6

[14] Baotian Hu, Qingcai Chen, and Fangze Zhu. Lcsts: A large scale chinese short text summarization dataset. *arXiv preprint arXiv:1506.05865*, 2015. 3

[15] Hai Hu, Kyle Richardson, Liang Xu, Lu Li, Sandra Kübler, and Lawrence S Moss. Ocnli: Original chinese natural language inference. *arXiv preprint arXiv:2010.05444*, 2020. 4

[16] Xuanjing Huang, Jing Jiang, Dongyan Zhao, Yansong Feng, and Yu Hong. *Natural language processing and Chinese computing: 6th CCF international conference, NLPCC 2017, Dalian, China, November 8–12, 2017, Proceedings*, volume 10619. Springer, 2018. 4

[17] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. Curricularface: adaptive curriculum learning loss for deep face recognition. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5901–5910, 2020. 6

[18] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021. 1

[19] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020. 1

[20] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Learning image representations by completing damaged jigsaw puzzles. In *IEEE Winter Conference on Applications of Computer Vision*, pages 793–802. IEEE, 2018. 3

[21] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2020. 3

[22] Peng Li, Wei Li, Zhengyan He, Xuguang Wang, Ying Cao, Jie Zhou, and Wei Xu. Dataset and neural recurrent sequence labeling model for open-domain factoid question answering. *arXiv preprint arXiv:1607.06275*, 2016. 4

[23] Yudong Li, Yuqing Zhang, Zhe Zhao, Linlin Shen, Weijie Liu, Weiquan Mao, and Hui Zhang. Csl: A large-scale chinese scientific literature dataset. *arXiv preprint arXiv:2209.05034*, 2022. 3

[24] Jimmy Lin. A proposed conceptual framework for a representational approach to information retrieval. In *ACM SIGIR Forum*, volume 55, pages 1–29. ACM New York, NY, USA, 2022. 3

[25] Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th international conference on computational linguistics*, pages 1952–1962, 2018. 4

[26] Zheng Liu and Yingxia Shao. Retromae: Pre-training retrieval-oriented transformers via masked auto-encoder. *arXiv preprint arXiv:2205.12035*, 2022. 5

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[28] Yuxiang Lu, Yiding Liu, Jiaxiang Liu, Yunsheng Shi, Zhengjie Huang, Shikun Feng Yu Sun, Hao Tian, Hua Wu, Shuaiqiang Wang, Dawei Yin, et al. Ernie-search: bridging cross-encoder with dual-encoder via self onthe-fly distillation for dense passage retrieval. *arXiv preprint arXiv:2205.09153*, 2022. 1

[29] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021. 1

[30] A Neelakantan, T Xu, R Puri, A Radford, JM Han, J Tworek, Q Yuan, N Tezak, JW Kim, C Hallacy, et al. Text and code embeddings by contrastive pre-training. arxiv 2022. *arXiv preprint arXiv:2201.10005*, 2022. 1

[31] OpenAI. Gpt-4 technical report, 2023. 1, 4

[32] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022. 1

[33] Yifu Qiu, Hongyu Li, Yingqi Qu, Ying Chen, Qiaoqiao She, Jing Liu, Hua Wu, and Haifeng Wang. Dureader_retrieval: A large-scale chinese benchmark for passage retrieval from web search engine. *arXiv preprint arXiv:2203.10232*, 2022. 4, 6

[34] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191*, 2020. 1

[35] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019. 3

[36] Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. Pair: Leveraging passage-centric similarity relation for improving dense passage retrieval. *arXiv preprint arXiv:2108.06027*, 2021. 1

[37] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. 3

[38] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995. 3

[39] Chih Chieh Shao, Trois Liu, Yuting Lai, Yiying Tseng, and Sam Tsai. Drcd: A chinese machine reading comprehension dataset. *arXiv preprint arXiv:1806.00920*, 2018. 4

[40] M Sun, J Li, Z Guo, Z Yu, Y Zheng, X Si, and Z Liu. Thuctc: An efficient chinese text classifier. *URL https://github. com/thunlp/THUCTC*, 2016. 4

[41] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*, 2021. 1

[42] Hongxuan Tang, Hongyu Li, Jing Liu, Yu Hong, Hua Wu, and Haifeng Wang. Dureader_robust: A chinese dataset towards evaluating robustness and generalization of machine reading comprehension in real-world applications. *arXiv preprint arXiv:2004.11142*, 2020. 4

[43] Hongyin Tang, Xingwu Sun, Beihong Jin, Jingang Wang, Fuzheng Zhang, and Wei Wu. Improving document representations by generating pseudo query embeddings for dense retrieval. *arXiv preprint arXiv:2105.03599*, 2021. 3

[44] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[45] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Douglas Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Kinney, et al. Cord-19: The covid-19 open research dataset. *ArXiv*, 2020. 4

[46] BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022. 1

[47] Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Tianyang Zhang, Xianpei Han, Zhen Hu, Heng Wang, et al. Cail2019-scm: A dataset of similar case matching in legal domain. *arXiv preprint arXiv:1911.08962*, 2019. 4

[48] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*, 2023. 1, 2, 4, 6

[49] Xiaohui Xie, Qian Dong, Bingning Wang, Feiyang Lv, Ting Yao, Weinan Gan, Zhijing Wu, Xiangsheng Li, Haitao Li, Yiqun Liu, et al. T2ranking: A large-scale chinese benchmark for passage ranking. *arXiv preprint arXiv:2304.03679*, 2023. 4

[50] Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, et al. Clue: A chinese language understanding evaluation benchmark. *arXiv preprint arXiv:2004.05986*, 2020. 4

[51] Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. Paws-x: A cross-lingual adversarial dataset for paraphrase identification. *arXiv preprint arXiv:1908.11828*, 2019. 4

[52] Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. Pretrained transformers for text ranking: Bert and beyond. In *Proceedings of the 14th ACM International Conference on web search and data mining*, pages 1154–1156, 2021. 3

[53] Sha Yuan, Hanyu Zhao, Zhengxiao Du, Ming Ding, Xiao Liu, Yukuo Cen, Xu Zou, Zhilin Yang, and Jie Tang. Wudaocorpora: A super large-scale chinese corpora for pre-training language models. *AI Open*, 2:65–68, 2021. 3

[54] Bowei Zhang, Weiwei Sun, Xiaojun Wan, and Zongming Guo. Pku paraphrase bank: A sentence-level paraphrase corpus for chinese. In *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part I 8*, pages 814–826. Springer, 2019. 4

[55] Jiangtao Zhang, Juanzi Li, Zengtao Jiao, and Jun Yan. Overview of ccks 2018 task 1: named entity recognition in chinese electronic medical records. In *Knowledge Graph and Semantic Computing: Knowledge Computing and Language Understanding: 4th China Conference, CCKS 2019, Hangzhou, China, August 24–27, 2019, Revised Selected Papers 4*, pages 158–164. Springer, 2019. 4

[56] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 1

A Appendix / supplemental material

Optionally include supplemental material (complete proofs, additional experiments and plots) in the appendix. All such materials **SHOULD be included in the main submission.**

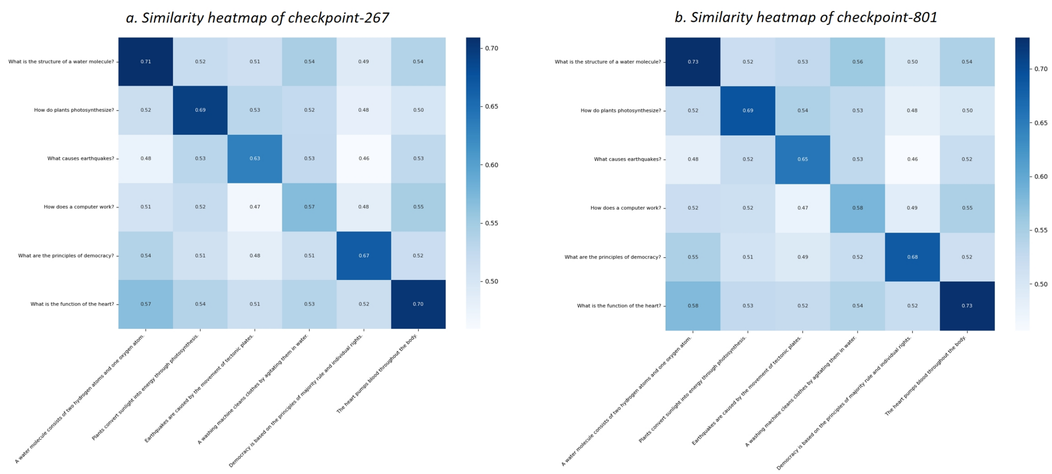

Figure 5: Visualization of similarity heatmap from different checkpoints of the PEG method

A.1 The impact of batch size and train group size

We investigated the influence of both batch size and training group size. During the data construction phase, we extracted numerous challenging samples for each query and supplemented these with our own positive samples to establish the training group size. The corresponding results are presented in Table 7. We found that a reduction in batch size significantly impairs model performance. Similarly, the size of the training group has a substantial effect on the model's performance.

Table 7: The impact of batch size.

Batch Size	Train Group Size		
	1	3	6
3K	60.63	61.42	61.26
6K	64.08	64.21	64.55
13K	65.28	66.27	**66.33**

A.2 Usage of Large Language Models in Data Processing

Generation of Queries by LLMs: To generate queries for our privately collected corpus, which only contains raw texts, we employed an off-the-shelf LLM (GPT3.5 api). The LLM was used to generate questions and answers related to each passage, which were chunked from long texts. The process involved the following prompt:

Below is an instruction that describes a task. Write a response that appropriately completes the request. Instruction: Generate 5 questions and answers related to the content of the following passage. - The questions generated need to be able to find answers from the passage - The result is returned in json format: {"qas": [{"question": "Generated question 1", "answer": "Answer to question 1"}, {"question": "Generated question 2", "answer": "Answer to question 2"}]} Passage: passage Response:

This prompt instructs the LLM to generate five questions and their corresponding answers for each passage, ensuring that the questions are relevant and answerable from the passage content.

Relevance Filtering by LLMs: Despite the initial pairing, the collected query-passage pairs still contained false positives and negatives, with an observed noise proportion of approximately 8.2%.

To improve the quality of these pairs, we used the LLM to classify the relevance of the paired queries and passages. The following prompt was used for this relevance filtering:

Query: {query} Passage: {passage} Given the query and the passage above, answer the question below: Is the query positively/negatively related to the passage?

A.3 Heatmap Analysis of the PEG

To more intuitively demonstrate the impact of the progressive learning method on the model's ability to compute text similarity at different training stages, we visualized the similarity heatmaps generated by the PEG method at various checkpoints, as shown in Figure 5.

The color intensity in the heatmaps indicates the degree of similarity, with darker colors representing higher similarity scores. Unlike traditional similarity computation methods, such as cosine similarity, which produce static similarity values for the same samples throughout the training process, the PEG method introduces dynamic weight adjustment. This allows the model to produce different similarity values for the same positive and negative sample pairs at different training stages. This dynamic adjustment significantly enhances the model's robustness in learning high-quality text embeddings, thereby improving its performance in text retrieval tasks.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction have accurately reflected the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The possible limitations have been described in chapter 5, conclusion.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: We use the progressively learning strategy to improve contrastive learning and conduct experimental verification. Both theories have been theoretically proven.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We will open source our code and data after the paper is published, and have opened the checkpoint of our model on Hugginface.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code and dataset will be released upon acceptance

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The training and test details have been described in chapter 4 and 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Described in Section 4.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Described in section 4.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We strictly abide by the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The contribution of the article has been emphasised in the Abstract, in the Introduction section of Chapter 1 and in the Conclusion section of Chapter 5, and the possible limitations of the article have been described.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We explain our data sources in detail in Section 3.2.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All references to the use of other people's codes, data, models are referenced by links.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: Code and dataset will be released upon acceptance.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [No]

Justification: The experiments in this paper did not include any human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The experiments in this paper did not include any human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.