PEG: Towards Robust Text Retrieval with Progressive Learning

Anonymous ACL submission

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 are detrimental to the semantic correctness and consistency of embeddings. Third, the equal treatment to 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.

1 Introduction

Information (knowledge) retrieval, a crucial aspect of natural language processing, plays an increasing role in the context of large language models (LLMs) (Ouyang et al., 2022; Nakano et al., 2021; OpenAI, 2023; Touvron et al., 2023; Chowdhery et al., 2022; Workshop et al., 2022; Zhang et al., 2022; Sun et al., 2021). 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) (Gao and Callan, 2021; Qu et al., 2020; Izacard et al., 2021; Gao et al., 2021; Ren et al., 2021; Lu et al., 2022; Khattab and Zaharia, 2020). Its steps are composed of the text encoding and text matching, where 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 similarity between queries and document fragments are 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 (Gao et al., 2021; Izacard et al., 2021; Neelakantan et al., 2022; Xiao et al., 2023). 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, method such as SimCSE (Gao et al., 2021) employs 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 are used to 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 contrastive learn-

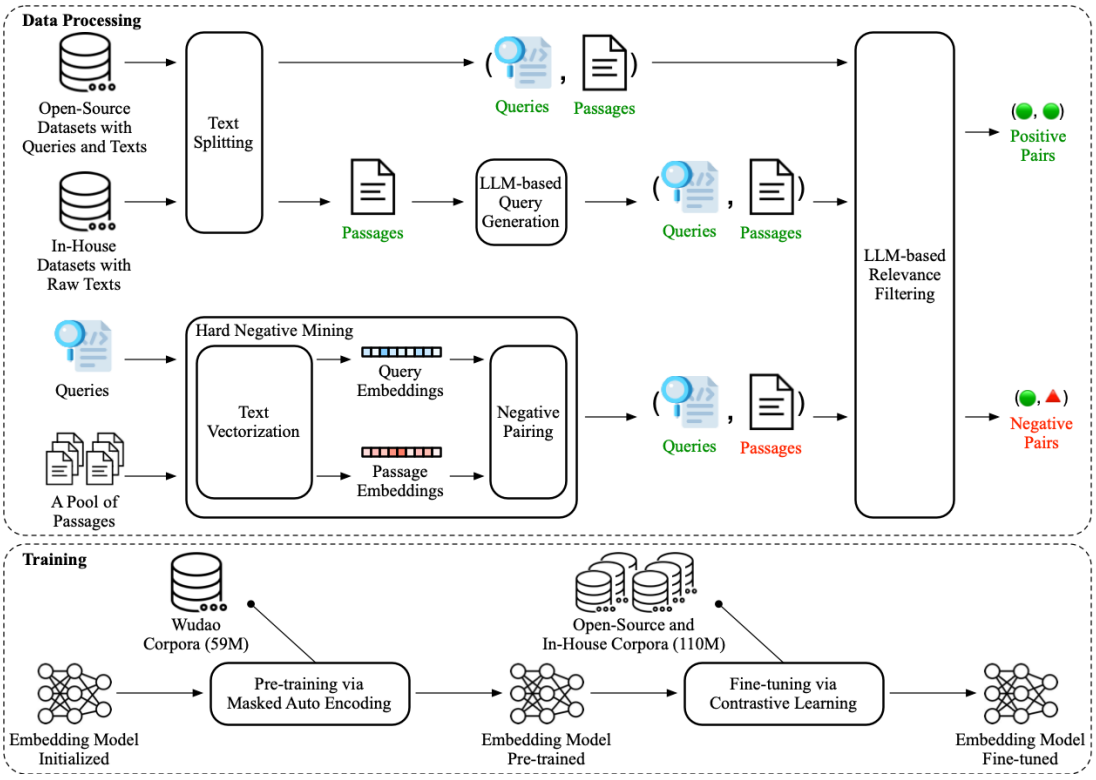

Figure 1: The pipeline of PEG. During data processing, we first split raw texts into shorter passages to control the length of text segments from different datasets. Since our in-house datasets only contain passages without the structured queries, we perform query generation based on each passage via a 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 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.

ing based embedding learning is that the representation capacity is closely related to the quality and quantity of negative samples. A small batch size containing insufficient high-quality, diversified negative samples may not effectively compel the model to discern the subtle differences among highly similar samples, thus impeding its ability to achieve superior discrimination. Consequently, BGE (Xiao et al., 2023) 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, disregarding the varying difficulties of learning easy and hard negatives. The overfitting of a majority of simple negatives ultimately leads to sub-optimal convergence.

In order 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 downstreaming 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 a 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 negative. 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 with respect to 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:

- 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.
- We propose PEG, which progressively adjust the weights of samples based on the difficulties of negative samples.
- Extensive experiments demonstrate that PEG achieves the SOTA performance.

2 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).

2.1 Dataset Source

Pre-training. We make use of the publicly available Wudao Corpora (Yuan et al., 2021) 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 dataset 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 (Yuan et al., 2021), LCSTS (Hu et al., 2015), WeiXin Public Corpus[1], CSL (Li et al., 2022), NLPCC 2017 (Huang et al., 2018), DRCD (Shao et al., 2018), and THUCNews (Sun et al., 2016). For the QA task, we utilize datasets like DuReader-Retrieval (Qiu et al., 2022), WebQA (Li et al., 2016), T2Ranking (Xie et al., 2023), mMARCO (Bonifacio et al., 2021), Chinese Medical Dialogue (Chen et al., 2020a), BaiDuZhidao[2], DuReader-Robust (Tang et al., 2020), and DuReader-Checklist[3]. For the text matching task, we use CMNLI (Xu et al., 2020), OCNLI (Hu et al., 2020), LCQMC (Liu et al., 2018), PAWSX (Yang et al., 2019), CCKS2018 (Zhang et al., 2019b), COVID19 (Wang et al., 2020), ChineseSTS[4], CMRC (Cui et al., 2019), AdvertiseGen[5], ATEC[6], BQ[7], GAIIC2021-OPPO[8], CAIL2019-SCM (Xiao et al., 2019), and PKU-Paraphrase-Bank (Zhang et al., 2019a). Our in-house datasets are primarily composed of high-quality books and journals, covering domains of finance, medicine, tourism, laws and policies, and logistics.

2.2 Data Processing

Pre-training. We follow BGE (Xiao et al., 2023) 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 (OpenAI, 2023)) 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

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

[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

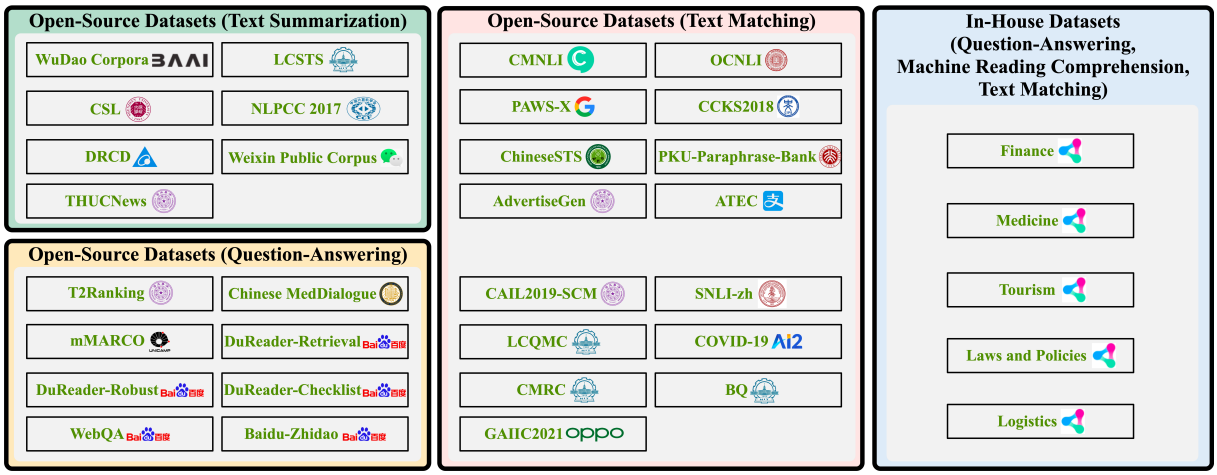

Figure 2: Overview of the datasets. Both open-source and in-house datasets contain a variety of tasks and domains.

(*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 a 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.

2.3 Training

Pre-Training. Our model is pre-trained on the Wudao corpora. We utilize the MAE-style approach, as presented in RetroMAE (Liu and Shao, 2022), 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 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(He et al., 2020) for model optimization:

$$\mathcal{L}_{ft} = \sum_{(\boldsymbol{e}_q, \boldsymbol{e}_p)} -\log \frac{h(\boldsymbol{e}_q, \boldsymbol{e}_p)}{h(\boldsymbol{e}_q, \boldsymbol{e}_p) + \sum_n^N h(\boldsymbol{e}_q, \boldsymbol{e}_n)},$$

$$\begin{aligned} h(\boldsymbol{e}_q, \boldsymbol{e}_p) &= \exp(\mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_p)/\tau), \\ h(\boldsymbol{e}_q, \boldsymbol{e}_n) &= \exp(\mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_n)/\tau), \end{aligned} \tag{2}$$

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 circumstance, 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

[9]https://huggingface.co/shibing624/text2vec-base-chinese

focus on simple samples in the initial stages to first gain the preliminary knowledge. Then, it gradually shift 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:

$$\mathcal{L}_{ft} = \sum_{(\boldsymbol{e}_q, \boldsymbol{e}_p)} -w_q \log \frac{h(\boldsymbol{e}_q, \boldsymbol{e}_p)}{h(\boldsymbol{e}_q, \boldsymbol{e}_p) + \sum_n^N g(a_n, \boldsymbol{e}_q, \boldsymbol{e}_n)}$$

$$g(a_n, \boldsymbol{e}_q, \boldsymbol{e}_n) = \exp(a_n \cdot \mathrm{s}(\boldsymbol{e}_q, \boldsymbol{e}_n)/\tau), \tag{3}$$

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

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

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

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 $\boldsymbol{e}_q$ and $\boldsymbol{e}_p$. 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 (He et al., 2020; Huang et al., 2020), 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.

3 Experiments

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

3.1 Datasets

C-MTEB. The Chinese Massive Text Embedding Benchmark (C-MTEB) (Xiao et al., 2023) 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 encompass 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 (Qiu et al., 2022) contains training set, development set, 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

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**

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.

3.2 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).

3.3 Implementation details

We use BERT-large (Devlin et al., 2018) model as our basic model architecture. We train our model on 32 H800 GPUs. For pre-training, we use AdamW (Loshchilov and Hutter, 2017) 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 $\tau = 0.01$.

3.4 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 exsiting 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.

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**	**84.8**	**85.14**	**67.72**

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,

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**

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

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 around 4% in Recall@1 and 2% in MRR@10. It's worth noting that the size 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.

3.5 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.

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.

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)

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.

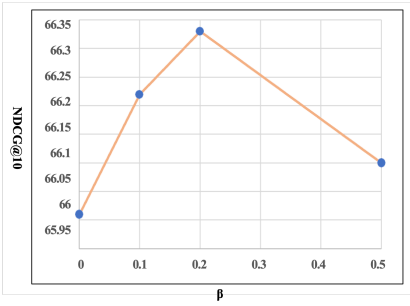

Figure 3: The impact of hyperparameter β.

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).

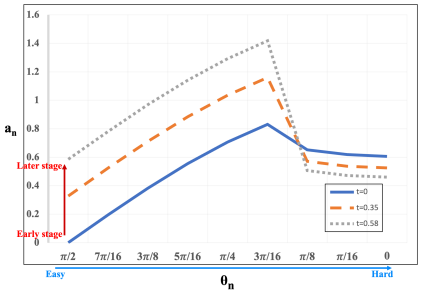

Figure 4: The trend of negative sample weights.

4 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 (Robertson et al., 1995, 2009), lexical matching is performed while for the dense retrieval like Condenser (Gao and Callan, 2021), 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 (Yates et al., 2021; Fan et al., 2022; Guo et al., 2022; Lin, 2022). REALM (Guu et al., 2020) 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 (Tang et al., 2021) generated multiple pseudo query embeddings for representation of documents and boost the retrieval performance via the nearest neighbor search. TASER (Cheng et al., 2022) 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 (Kim et al., 2018); 2) instance-wise contrast, where similar image pairs are pulled closer with dissimilar pairs pushed farther (He et al., 2020; Chen et al., 2020b). PCL (Li et al., 2020) encourages each image embedding to be adjacent to its assigned cluster prototype for unsupervised contrastive representation learning. SentenceBERT (Reimers and Gurevych, 2019) explicitly learns a sentence embedding using the triplet loss where two sentences from the same passage are considered positive pairs and are negative otherwise.

5 Conclusion

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.

6 Limitations

In the future, we will carry out evaluations on datasets encompassing a broader range of languages.

References

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

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

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

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

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

Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. 2019. 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.

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

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

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. *arXiv preprint arXiv:2104.08253*.

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

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

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

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

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

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

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

Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. 2020. 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.

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*.

Omar Khattab and Matei Zaharia. 2020. 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.

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

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

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

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

Jimmy Lin. 2022. 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.

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

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

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

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

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

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

OpenAI. 2023. Gpt-4 technical report.

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

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

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

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

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

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

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

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

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

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

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

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

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

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

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

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. *arXiv preprint arXiv:2309.07597*.

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

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

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

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. 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.

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

Bowei Zhang, Weiwei Sun, Xiaojun Wan, and Zongming Guo. 2019a. 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.

Jiangtao Zhang, Juanzi Li, Zengtao Jiao, and Jun Yan. 2019b. 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.

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

A 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 6. 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 6: 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**