C-Pack: Packed Resources For General Chinese Embeddings

Shitao Xiao[†]
stxiao@baai.ac.cn
Beijing Academy of AI
Beijing, China

Zheng Liu[†*]
zhengliu1026@gmail.com
Beijing Academy of AI
Beijing, China

Peitian Zhang
namespace.pt@gmail.com
Renmin University of China
Beijing, China

Niklas Muennighoff
n.muennighoff@gmail.com
HuggingFace
Beijing, China

Defu Lian
liandefu@ustc.edu.cn
USTC
Hefei, China

Jian-Yun Nie
nie@iro.umontreal.ca
University of Montreal
Montreal, Canada

ABSTRACT

We introduce **C-Pack**, a package of resources that significantly advances the field of general text embeddings for Chinese. **C-Pack** includes three critical resources. 1) **C-MTP** is a massive training dataset for text embedding, which is based on the curation of vast unlabeled corpora and the integration of high-quality labeled corpora. 2) **C-MTEB** is a comprehensive benchmark for Chinese text embeddings covering 6 tasks and 35 datasets. 3) **BGE** is a family of embedding models covering multiple sizes. Our models outperform all prior Chinese text embeddings on **C-MTEB** by more than +10% upon the time of the release. We also integrate and optimize the entire suite of training methods for **BGE**. Along with our resources on general Chinese embedding, we release our data and models for English text embeddings. The English models also achieve state-of-the-art performance on the MTEB benchmark; meanwhile, our released English data is 2 times larger than the Chinese data. Both Chinese and English datasets are the largest public release of training data for text embeddings. All these resources are made publicly available at https://github.com/FlagOpen/FlagEmbedding.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**.

KEYWORDS

Text Embeddings, Training Data, Benchmark, Pre-trained Models

ACM Reference Format:
Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed Resources For General Chinese Embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3626772.3657878

[†]These two researchers are co-first authors.
[*]Zheng Liu is the corresponding author

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657878

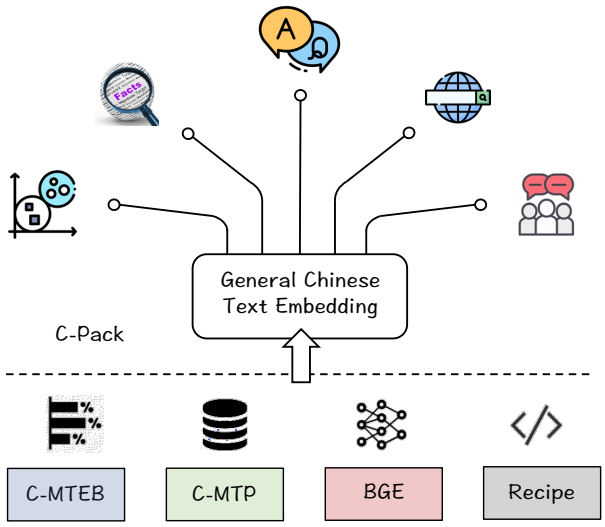

Figure 1: C-Pack presents 4 critical resources to support general Chinese embedding: C-MTEB (comprehensive evaluation benchmark), C-MTP (massive training data), BGE (powerful pre-trained models), the entire-suite of training recipe.

1 INTRODUCTION

Text embedding is a long-standing topic in natural language processing and information retrieval. By representing texts with latent semantic vectors, text embedding can support various applications, e.g., web search, question answering, and retrieval-augmented language modeling [26, 56, 59, 64]. The recent popularity of large language models (LLMs) has made text embeddings even more important. Due to the inherent limitations of LLMs, such as world knowledge and action space, external support via knowledge bases or tool use is necessary. Text embeddings are critical to connect LLMs with these external modules [9, 41].

The wide variety of application scenarios calls for a single unified embedding model that can handle all kinds of usages (like retrieval, ranking, classification) in any application scenarios (e.g., question answering, language modeling, conversation). However, learning general-purpose text embeddings is much more challenging than task-specific ones. The following factors are critical:

• **Data**. The development of general-purpose text embeddings puts forward much higher demands on the training data in terms of

scale, *diversity*, and *quality*. To achieve high discriminative power for the embeddings, it may take more than hundreds of millions of training instances [22, 40, 53], which is orders of magnitude greater than typical task-specific datasets, like MS MARCO [38] and NLI [10, 55]. Besides scale, the training data needs to be collected from a wide range of sources so as to improve the generality across different tasks [22, 53]. Finally, the augmentation of scale and diversity will probably introduce noise. Thus, the collected data must be properly cleaned before being utilized for the training of embeddings [53].

• **Training**. The training of general-purpose text embeddings depends on two critical elements: a well-suited backbone encoder and an appropriate training recipe. While one can resort to generic pre-trained models like BERT [15] and T5 [45], the quality of text embedding can be substantially improved by pre-training with large-scale unlabeled data [22, 53]. Further, instead of relying on a single algorithm, it takes a compound recipe to train general-purpose text embedding. Particularly, it needs embedding-oriented pre-training to prepare the text encoder [16], contrastive learning with sophisticated negative sampling to improve the embedding's discriminability [43], and instruction-based fine-tuning [7, 50] to integrate different representation capabilities of text embedding.

• **Benchmark**. Another pre-requisite condition is the establishment of proper benchmarks, where all needed capabilities of text embeddings can be comprehensively evaluated. BEIR [51] provides a collection of 18 to evaluate the embedding's general performances on different retrieval tasks, e.g., question answering and fact-checking. Later, MTEB [35] proposes a more holistic evaluation of embeddings and extends BEIR. It integrates 56 datasets, where all important capabilities of text embeddings, like retrieval, ranking, clustering, etc., can be jointly evaluated.

Altogether, the development of general-purpose text embedding needs to be made on top of a mixture of driving forces, from data, and encoder models, to training methods and benchmarking. In recent years, continual progresses have been achieved in this field, such as Contriever [22], E5 [53], GTR [40], and OpenAI Text Embedding [37]. Nevertheless, most of these models are dedicated to the English-centric scenarios. In contrast, there is a shortage of competitive models for general Chinese embedding. What is worse, the development of general Chinese embedding is severely constrained in many aspects: there are neither well-prepared training resources nor suitable benchmarks to evaluate the generality.[1]

To address the above challenges, we present a package of resources called **C-Pack**, which contributes to the development of general Chinese embedding from the following perspectives.

• **C-MTEB** (Chinese Massive Text Embedding Benchmark). The benchmark is established as a Chinese extension of MTEB.[2] **C-MTEB** collects 35 public-available datasets belonging to 6 types of tasks. We set up the unified testing protocols so that different embeddings can be evaluated on fair ground. We also develop the evaluation pipeline which significantly makes ease for the evaluation process. Thanks to the scale and diversity of **C-MTEB**, all major capabilities of Chinese embeddings can be reliably measured, making it the most suitable benchmark to evaluate the generality of Chinese text embedding.

• **C-MTP** (Chinese Massive Text Pairs). We create a massive training dataset of 100M text pairs. The majority of our dataset is curated from the massive web corpora, such as Baike (Wikipedia-style webs in Chinese), Zhihu (a major Chinese social media), major News Websites in Chinese. We extract the semantically related text pairs leveraging the rich-structured information within the data, such as title-to-document, subtitle-to-passage, question-to-answer, question-to-similar-question, etc. The extracted data is further cleaned for the massive weakly supervised training of the text embeddings. We also integrate diverse labeled datasets, which presents high-quality supervision signals for the final refinement of text embeddings. Besides, considering that there is no public available dataset for general English text embeddings either, we curate another massive dataset for English with the same method, which consists of 200M text pairs.

• **BGE** (BAAI General Embeddings). We provide a family of well-trained models for Chinese general text embeddings. There are three optional model sizes: small (24M), base (102M), and large (326M), which present users with the flexibility to trade off efficiency and effectiveness. Our models make a big leap forward in generality: **BGE** outperforms all previously Chinese text embedding models on all aspects of **C-MTEB** by large margins. Besides being directly applicable, **BGE** can also be fine-tuned with additional data for better downstream performances. The releasing of these powerful models substantially contributes to critical applications, such as search, question answering, and retrieval-augmented generation.

• **Training Recipe**. Accompanying our resources, we integrate and optimize training methods to build general-purpose text embeddings, including the pre-training of an embedding-oriented text encoder, general-purpose contrastive learning, and task-specific fine-tuning. The release of the training recipe will help the community to reproduce the state-of-the-art methods and make continuous progress on top of them.

Our project enjoys a widespread popularity in technique communities, like HuggingFace and Github. Remarkably, according to the up-to-date statics (2024-04), BGE model series have received more than **20 million downloads** from HuggingFace since its release on 2023-08, making it one of the most popular embedding models in the world. They have been integrated by the major RAG and text-embedding frameworks in the world, such as Langchain[3], LLamaIndex[4], and Huggingface[5]. The code-base also receives nearly **5,000 stars** on GitHub. So far, there have been over 100 submissions on C-MTEB, and it is widely recognized as the **most popular and authoritative benchmark** for Chinese text embeddings. It's worth noting that our project is still fast growing, with new resources continually created and released to the public. In summary, C-Pack provides a go-to option for the **development**, **evaluation**, and **application** of general-purpose Chinese text embedding, which establishes a solid foundation for the advancement of this field.

The remaining part of this paper is organized as follows. We discuss the related works about general text embeddings in Section 2. We present a detailed introduction about C-Pack resources in Section 3. We make comprehensive empirical analysis for the value of C-Pack in Section 4. Finally, we conclude our work in Section **??**.

[1] This situation has been substantially improved since our work. New methods are continually developed on top of the benchmark, data, and models from C-Pack.
[2] https://huggingface.co/spaces/mteb/leaderboard
[3] https://python.langchain.com/docs/integrations/text_embedding/bge_huggingface
[4] https://docs.llamaindex.ai/en/stable/examples/embeddings/huggingface.html
[5] https://huggingface.co/docs/text-embeddings-inference/index

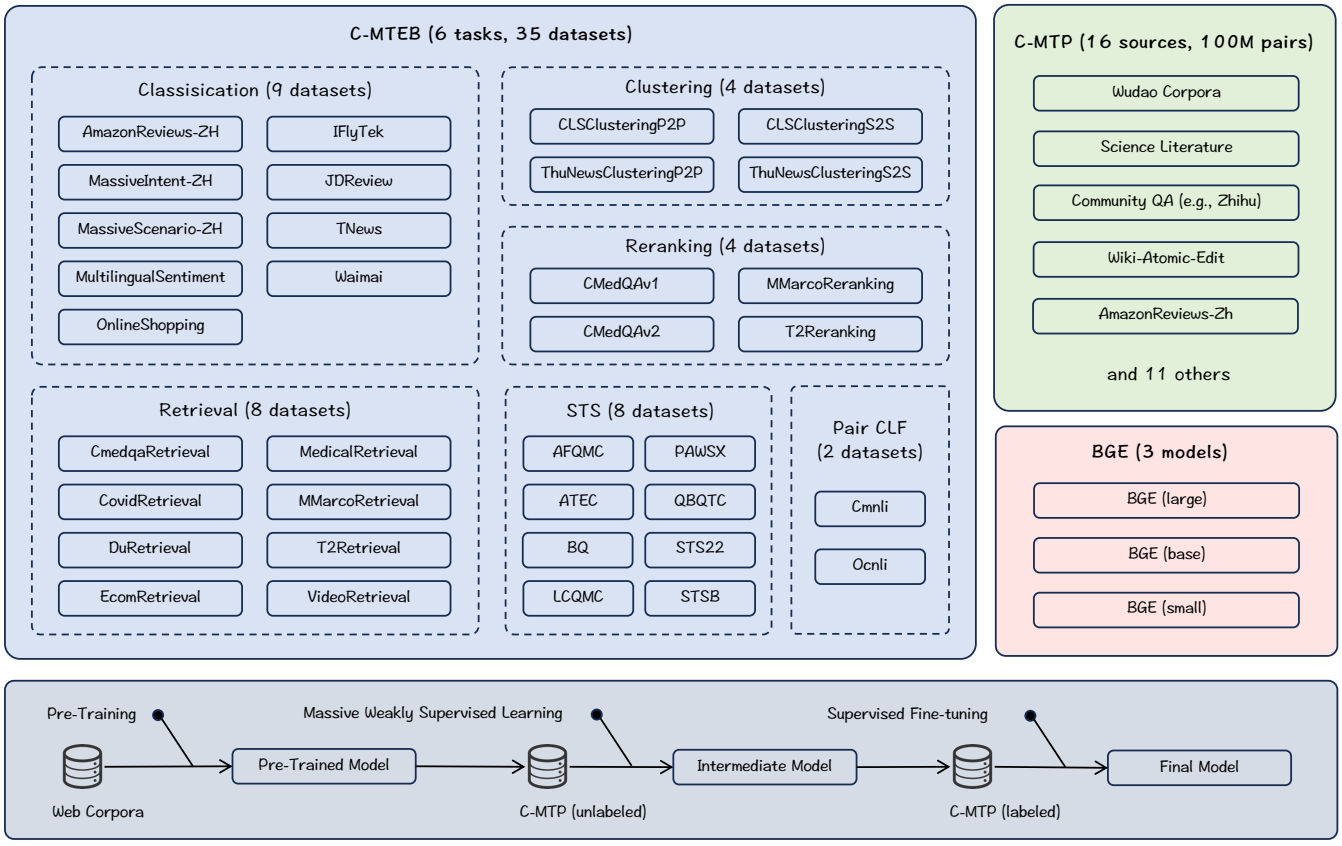

Figure 2: Overview of C-Pack. C-MTEB is a benchmark for Chinese text embeddings. C-MTP is a large-scale Chinese embedding training dataset. BGE are state-of-the-art Chinese embedding models. The training recipe is shown at the bottom.

2 RELATED WORK

The importance of general text embedding is widely recognized, not only for its wide usage in typical applications, like web search and question answering [25, 57] but also due to its fundamental role in augmenting large language models [9, 19, 23, 26, 48]. Compared with the conventional task-specific methods, the general text embedding needs to be extensively applicable in different scenarios. In recent years, there has been a continual effort in this field, where a series of well-known works are proposed, like Contriever [22], GTR [40], sentence-T5 [39], Sentence-Transformer [46], E5 [52], OpenAI text embedding [37], etc. Although it remains an open problem, recent studies highlight the following important factors.

• Firstly, the training data is desired to be large-scale and diversified, from which the embedding model can learn to recognize different kinds of semantic relationships [22, 37, 40, 53]. It usually calls for the comprehensive and elaborate data curation from web corpora, such as online encyclopedia, QA platforms, news websites, and social media communities. Despite similar efforts made by the previous works, few of the curated datasets are made public available before C-Pack.

• Secondly, the embedding model must be scaled up in terms of both training and model size, as scaled-up text encoders are more generalizable across different application scenarios [32, 39, 40]. Such an observation is in line with the conclusion for the importance of scaling LLMs [6, 11, 12, 17, 21, 27, 34, 44, 49]. Although large-scale training is affordable for many industrial organizations and companies, it remains a huge burden for the community users. The public release of BGE saves such a cost, benefiting both direct applications of text embeddings and further improvements for specific scenarios.

• Thirdly, the training recipe must be optimized through pre-training [30, 52], negative sampling [22, 52], and multi-task fine-tuning [7, 13, 33, 36, 47, 50, 54]. In C-Pack, these operations are integrated, optimized, and pipelined, which significantly facilitates people's reproduction and continual fine-tuning of BGE.

• Aside from the above factors, it is also critical to establish proper benchmarks to evaluate the generality of text embeddings. Unlike previous task-specific evaluations, like MSMARCO [38], SentEval [14], it is needed to substantially augment the benchmarks so as to evaluate the embedding's performance for a wide variety of tasks. One representative work is made by BEIR [24, 51], where the embeddings can be evaluated across different retrieval tasks. It is later extended by MTEB [35], where all major aspects of text embeddings can be comprehensively evaluated. However, no such works were done for the Chinese community before. By introducing C-MTEB, this limitation has been substantially conquered.

Given the above analysis, it can be concluded that the general text embedding is highly resource-dependent, which calls for a wide range of elements, such as datasets, models, and benchmarks. Thus, the creation and public release of the corresponding resources in C-Pack is crucially important.

3 C-PACK

In this section, we first introduce the resources in C-Pack: the benchmark **C-MTEB**, the training data **C-MTP**, and the model class **BGE**. Then, we discuss the training recipe, which enables us to train the state-of-the-art models for general Chinese embedding based on the offered resources.

3.1 Benchmark: C-MTEB

C-MTEB is established for the comprehensive evaluation of the generality of Chinese embeddings (Figure 2). In the past few years, the community has put forward many datasets for text representation and language understanding tasks in Chinese, such as CMNLI [62], DuReader [20], T^{2}Ranking [60]. However, these datasets are independently curated, lacking a fair and shared ground to comprehensively evaluate the general capability of text embeddings. Therefore, we create **C-MTEB**, where the following important efforts are made: 1) the comprehensive collection of datasets which can be either directly utilized or repurposed for the evaluation of text embeddings, 2) the categorization of the datasets into different capability attributes of text embeddings, e.g., retrieval, similarity analysis, classification, etc., 3) the standardization of the evaluation protocols, 4) the establishment of evaluation pipelines.

In particular, we collect a total of 35 public datasets related to the evaluation of Chinese text embeddings (Briefed as Figure 1. Detailed specifications are presented in the Github Repository[6]). The collected datasets are categorized based on the embedding's capability they may evaluate. There are 6 groups of evaluation tasks: retrieval, re-ranking, STS (semantic textual similarity), classification, pair classification, and clustering, which cover the main interesting aspects of Chinese text embeddings. Note that there are multiple datasets for each category. The datasets of the same category are collected from different domains and complementary to each other, therefore ensuring the corresponding capability to be fully evaluated.

The nature of each evaluation task and the evaluation metric are briefly introduced as follows.

• **Retrieval**. The retrieval task is presented with the test queries and a large corpus. For each query, it finds the Top-k similar documents within the corpus. The retrieval quality can be measured by ranking and recall metrics at different cut-offs. In this work, we use the setting from BEIR [51], using NDCG@10 as the main metric.

• **Re-ranking**. The re-ranking task is presented with test queries and their candidate documents (1 positive plus N negative documents). For each query, it re-ranks the documents based on the embedding similarity. The MAP score is used as the main metric.

• **STS** (Semantic Textual Similarity). The STS [1–5] task is to measure the correlation of two sentences based on their embedding similarity. Following the original setting in Sentence-BERT [46], the Spearman's correlation is computed with the given label, whose result is used as the main metric.

• **Classification**. The classification task re-uses the logistic regression classifier from MTEB [35], where the average precision is used as the main metric.

• **Pair-classification**. This task deals with a pair of input sentences, whose relationship is presented by a binarized label. The relationship is predicted by embedding similarity, where the average precision is used as the main metric.

• **Clustering**. The clustering task is to group sentences into meaningful clusters. Following the original setting in MTEB [35], it uses the mini-batch k-means method for the evaluation, with batch size equal to 32 and k equal to the number of labels within the mini-batch. The V-measure score is used as the main metric.

Finally, the embedding's capability on each task is measured by the average performance of all datasets for that task. The embedding's overall generality is measured by the average performance of all datasets in **C-MTEB**. We set up the standardized and compact pipeline for all tasks where different embedding models can be evaluated on a fair basis (Figure 3). FlagDRESModel is the wapper for the custmoized embedding model, which implements the encoding method for query and document. ChineseTaskList is the task list of C-MTEB. The evaluation result is written to the output folder, which can be directly submitted to the C-MTEB leaderboard[7].

```python
## FlagDRESModel: wrapper of embedding model
## ChineseTaskList: task list of C-MTEB
from C_MTEB import FlagDRESModel, ChineseTaskList

## Load BGE as the running example
model = C_MTEB.FlagDRESModel(model_name_or_path='bge')

## Sequential evaluation and save to output folder
for task in C_MTEB.ChineseTaskList:
    evaluation = MTEB(tasks=[task])
    evaluation.run(model, output_folder=f"zh_results/")
```

Figure 3: The evaluation pipeline of C-MTEB.

3.2 Training Data: C-MTP

We curate the largest dataset **C-MTP** for the training of general Chinese embedding. The paired texts constitute the data foundation for the training of text embedding, e.g., a question and its answer, two paraphrase sentences, or two documents on the same topic. To ensure the generality of the text embedding, the paired texts need to be both large-scale and diversified. Therefore, **C-MTP** is collected from two sources. The majority of the data is based on the curation of massive unlabeled data, *a.k.a.* **C-MTP (unlabeled)**, which presents 100 millions of paired texts. Meanwhile, a small portion is from the comprehensive integration of high-quality labeled data, *a.k.a.* **C-MTP (labeled)**, which leads to about 1 million paired texts. The data collection process is briefly introduced as follows.

• **C-MTP (unlabeled)**. We look for a wide variety of corpora, where we can extract rich-semantic paired structures from the plain text, e.g., paraphrases, title-body. Our primary source of data comes from open web corpora. The most representative one is the

[6]https://github.com/FlagOpen/FlagEmbedding/tree/master/C_MTEB

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

Table 1: Composition of C-MTP

dataset	C-MTP (unlabeled)	C-MTP (labeled)
source	Wudao, Zhihu, Baike, CSL, XLSUM-Zh, Amazon-Review-Zh, CMRC, etc.	T^2-Ranking, mMARCO-Zh, DuReader, NLI-Zh, etc.
size	$100M$	$838K$

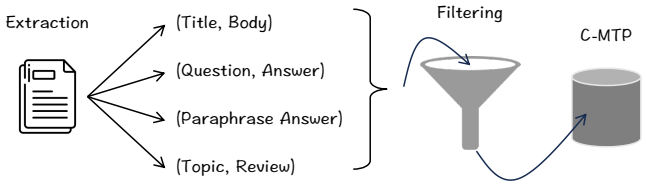

Figure 4: Creation of C-MTP.

Wudao corpus [63], which is the largest dataset of well-formatted articles for pre-training Chinese language models. For each its article, we extract structures, like (title, body), (sub-title, passage), to form a text pair. In addition, we collect data from other web content like Zhihu, Baike, news websites, which complement other forms of text pairs, especially (question, answer), (paraphrase titles), (paraphrase answers), etc. Aside from the open web content, we also explore other public Chinese data for more diverse text pairs, such as CSL (scientific literature), Amazon-Review-Zh (topic and its reviews), Wiki Atomic Edits (paraphrases), CMRC (machine reading comprehension), XLSUM-Zh (summarization), etc.

The text pairs curated from the web and other public sources are not guaranteed to be closely related. Therefore, data quality can be a major concern. In our work, we make use of a compound **data cleaning** strategy to refine the raw data. Firstly, the whole data undergoes general filtering, which removes non-textual, duplicated, and malicous content. Secondly, the data is further processed by semantic filtering so as to ensure the text pairs are semantically related. In our work, we make use a third-party model: Text2Vec-Chinese[8] to score the strength of relation for each text pair. We empirically choose a threshold of 0.43, and drop the samples whose scores are below the threshold. With such an operation, there are 100 million text pairs filtered from the unlabeled corpora. Despite the simplicity, we find that it effectively removes the irrelevant text pairs when manually reviewing samples and leads to strong empirical performances for the models trained on **C-MTP (unlabeled)**.

• **C-MTP (labeled)**. The labeled data is collected to further enhance to the training data. In our work, we integrate and re-purpose a diverse group of datasets, which covers different capabilities of the text embedding, like retrieval, ranking, similarity comparison, etc. Particularly, the following labeled datasets are included, T^2-Ranking [60], DuReader [20, 42], mMARCO [8], CMedQA-v2[65], multi-cpr[31], NLI-Zh[9], cmnli[62] and ocnli[62]. There are 838,465 paired texts in total, which contains diverse question-answering and paraphrasing patterns. Although it is much smaller than **C-MTP (unlabeled)**, most of the data is curated from human annotation, thus ensuring a high credibility of relevance.

Given the differences in scale and quality, **C-MTP (unlabeled)** and **C-MTP (labeled)** are applied to different training stages, which jointly result in a strong performance for the embedding model. Detailed analysis will be made in our training recipe.

3.3 Model Class: BGE

Even with the full package of training data, it is still challenging to learn general text embeddings due to the expensive training process. In our work, we provide a comprehensive class of well-trained embedding models for the community. Our models are based on the BERT-like architecture [15], which go through three-stage of training (to be discussed in the next section). There are three available scales: large (with 326M parameters), base (with 102M parameters), and small (with 24M parameters). The large-scale model achieves the highest general representation performances, leading the current public-available models by a considerable margin. The small-scale model is also empirically competitive compared with the public-available models and other model options in **BGE**; besides, it is way faster and lighter, making it suitable to handle massive knowledge bases and high-throughput applications. Thanks to the comprehensive coverage of different model sizes, people are presented with the flexibility to trade off running efficiency and representation quality based on their own needs.

As introduced, the models within **BGE** have been well-trained and achieve a strong generality for a wide variety of tasks. Meanwhile, they also establish a strong foundation for further fine-tuning. The fine-tuning recipe is well formulated in our training recipe, where all people's need is the preparation of fine-tuning data. It is empirically verified that the fine-tuned model may bring forth a much better performance for its application, compared with its original model in **BGE**, and the fine-tuned models from other general pre-trained encoders, like BERT. In other words, **BGE** not only presents people with direct usage embeddings but also works as a foundation where people may develop more powerful embeddings.

3.4 Training Recipe

The training recipe of **BGE** is completely released to the public along with C-Pack (Figure 2). Our training recipe has three main components: **1)** pre-training with plain texts, **2)** contrastive learning with **C-MTP (unlabeled)**, and **3)** multi-task learning with **C-MTP (labeled)**. As introduced, the public release of training recipe will not only help with the reproduction, but also benefit the improvement of BGE with continual training and fine-tuning.

• **Pre-Training**. Our model is pre-trained on massive plain texts through a tailored algorithm in order to better support the embedding task. Particularly, we make use of the Wudao corpora [63], which is a huge and high-quality dataset for Chinese language model pre-training. We leverage the MAE-style approach presented in RetroMAE [30, 58], which is simple but highly effective. The polluted text is encoded into its embedding, from which the clean text is recovered on top of a light-weight decoder:

$$\min . \sum_{x \in X} -\log \mathrm{Dec}(x|\mathbf{e}_{\tilde{X}}),\ \mathbf{e}_{\tilde{X}} \leftarrow \mathrm{Enc}(\tilde{X}).$$

[8] https://huggingface.co/GanymedeNil
[9] https://huggingface.co/datasets/shibing624/nli_zh

Table 2: Performance of various models on C-MTEB.

model	Dim	Retrieval	STS	Pair CLF	CLF	Re-rank	Cluster	Average
Text2Vec (base)	768	38.79	43.41	67.41	62.19	49.45	37.66	48.59
Text2Vec (large)	1024	41.94	44.97	70.86	60.66	49.16	30.02	48.56
Luotuo (large)	1024	44.40	42.79	66.62	61.0	49.25	44.39	50.12
M3E (base)	768	56.91	50.47	63.99	67.52	59.34	47.68	57.79
M3E (large)	1024	54.75	50.42	64.30	68.20	59.66	**48.88**	57.66
Multi. E5 (base)	768	61.63	46.49	67.07	65.35	54.35	40.68	56.21
Multi. E5 (large)	1024	63.66	48.44	69.89	67.34	56.00	48.23	58.84
OpenAI-Ada-002	1536	52.00	43.35	69.56	64.31	54.28	45.68	53.02
BGE (small)	512	63.07	49.45	70.35	63.64	61.48	45.09	58.28
BGE (base)	768	69.53	54.12	77.50	67.07	64.91	47.63	62.80
BGE (large)	1024	**71.53**	**54.98**	**78.94**	**68.32**	**65.11**	48.39	**63.96**

(Enc, Dec indicate the encoding and decoding operations, X, $\tilde{X}$ indicate the clean and polluted text.)

• **General purpose fine-tuning**. The pre-trained model is fine-tuned on **C-MTP (unlabeled)** via contrastive learning, where it is learned to discriminate the paired texts from their negative samples:

$$\min . \sum_{(p,q)} -\log \frac{e^{\langle \mathbf{e}_p, \mathbf{e}_q \rangle / \tau}}{e^{\langle \mathbf{e}_p, \mathbf{e}_q \rangle / \tau} + \sum_{Q'} e^{\langle \mathbf{e}_p, \mathbf{e}_{q'} \rangle / \tau}}.$$

(p and q are the paired texts, $q' \in Q'$ is a negative sample, τ is the temperature). One critical factor of contrastive learning is the negative samples. Instead of mining hard negative samples on purpose, we purely rely on in-batch negative samples [25] and resort to a big batch size (as large as 19,200) to improve the discriminativeness of the embedding.

• **Task-specific fine-tuning**. The embedding model is further fine-tuned with **C-MTP (labeled)**. The labeled datasets are smaller but of higher quality. However, the contained tasks are of different types, whose impacts can be mutually contradicted. In this place, we apply two strategies to mitigate this problem. On one hand, we leverage instruction-based fine-tuning [7, 50], where the input is differentiated to help the model accommodate different tasks. For each text pair (p, q), a task specific instruction I_t is attached to the query side: $q' \leftarrow q + I_t$. The instruction is a verbal prompt, which specifies the nature of the task, e.g., "*search relevant passages for the query*". On the other hand, the negative sampling is updated: in addition to the in-batch negative samples, one hard negative sample q' is mined for each text pair (p, q). The hard negative sample is mined from the task's original corpus, following the ANN-style sampling strategy in [61].

4 EXPERIMENTS

In this section, we conduct experimental studies for the exploration of following problems. **P1**. The extensive evaluation of different Chinese text embeddings on **C-MTEB**. **P2**. The empirical verification of the text embeddings by **BGE**. **P3**. The exploration of the practical value brought by **C-MTP**. **P4**. The exploration of the impacts introduced by the training recipe. We consider the following popular Chinese text embedding models as the baselines for our experiments: Text2Vec-Chinese[10] base and large; Luotuo[11]; M3E[12] base and large; multilingual E5 [53] and OpenAI text embedding ada 002[13]. The main metric presented in Section 3.1 is reported for each evaluation task in **C-MTEB**.

4.1 General Evaluation

We extensively evaluate **BGE** against popular Chinese text embeddings on **C-MTEB** as shown in Table 2.[14], where we can make the following observations.

First, our models outperform existing Chinese text embeddings by large margins. There is not only an overwhelming advantage in terms of the average performance, but also notable improvements for the majority of tasks in **C-MTEB**. The biggest improvements are on the retrieval task followed by STS, pair classification, and re-ranking. Such aspects are the most common functionalities of text embeddings, which are intensively utilized in applications like search engines, open-domain question answering, and the retrieval augmentation of large language models. Although the advantages for classification and clustering tasks are not as obvious, our performances are still on par or slightly better than the other most competitive models. The above observations verify the strong generality of **BGE**. *Our models can be directly utilized to support different types of application scenarios.*

Second, we observe performance growth resulting from the scaling up model size and embedding dimension. Particularly, the average performance improves from 58.28 to 63.96, when the embedding model is expanded from small to large. Besides the growth in average performance, there are also improvements across all the evaluation tasks. Compared to the other two baselines (Text2Vec, M3E), the impact of scaling up is more consistent and significant for our models. It is worth noting that our small model is still empirically competitive despite its highly reduced model size, where the average performance is even higher than the large-scale option of many existing models. As a result, *it provides people with the flexibility to trade-off embedding quality and running efficiency*: people may resort to our large-scale embedding model to deal with

[10] https://huggingface.co/shibing624
[11] https://huggingface.co/silk-road/luotuo-bert-medium
[12] https://huggingface.co/moka-ai
[13] https://platform.openai.com/docs/guides/embeddings
[14] Our **BGE** models are named **BGE** in the tables.

Table 3: Ablation of the training data, C-MTP, and the training recipe.

model	Dim	Retrieval	STS	Pair CLF	CLF	Re-rank	Cluster	Average
M3E (large)	1024	54.75	50.42	64.30	68.20	59.66	48.88	57.66
OpenAI-Ada-002	1536	52.00	43.35	69.56	64.31	54.28	45.68	53.02
BGE-*pretrain*	1024	63.90	47.71	61.67	**68.59**	60.12	47.73	59.00
BGE w.o. pre-train	1024	62.56	48.06	61.66	67.89	61.25	46.82	58.62
BGE w.o. Instruct	1024	70.55	53.00	76.77	68.58	64.91	**50.01**	63.40
BGE-*finetune*	1024	**71.53**	**54.98**	**78.94**	68.32	**65.11**	48.39	**63.96**

high-precision usages, or switch to the small-scale one for high-throughput scenarios.

Using the same recipe as the Chinese models, we also train a set of English text embedding models presented in Table 5. Besides, our English data was created and released together with C-MTP. It was the first time that such comprehensive training data was made publicly available. At the time of public release, our English BGE models achieved the state-of-the-art performance on the English MTEB benchmark [35] across its 56 datasets. Although there were many strong competitors in the English community, such as E5 [53], SGPT [32], GTE [28], GTR [40], and OpenAI Ada-002 [37], we were able to notably advance the prior SOTA by an absolute 1.1 points in total average, which further verify the effectiveness of our data curation and training method.

4.2 Detailed Analysis

We investigate the detailed impact of **C-MTP** and our **training recipe**. The corresponding experiment results are presented in Table 3 and Table 4, respectively.

First of all, we analyze the impact of our training data, **C-MTP**. As mentioned, **C-MTP** consists of two parts. 1) **C-MTP (unlabeled)**, which is used for general-purpose fine-tuning; the model produced from this stage is called the intermediate checkpoint, denoted as **BGE**-*pretrain*. 2) **C-MTP (labeled)**, where the task-specific fine-tuning is further conducted on top of BGE-*pretrain*; the model produced from this stage is called the final checkpoint, noted as **BGE**-*finetune*. Based on our observations from the experimental result, both **C-MTP (unlabeled)** and **C-MTP (labeled)** substantially contribute to the embedding's quality.

Regarding **C-MTP (unlabeled)**, despite mostly being curated from unlabeled corpora, this dataset alone brings forth strong empirical performance for the embedding models trained on it. Compared with other baselines like Text2Vec, M3E, and OpenAI text embedding, BGE-*pretrain* already achieves a higher average performance. A further look into the performances reveals more details. On one hand, **C-MTP (unlabeled)** makes a major impact on the embedding's retrieval quality, where BGE-*pretrain* notably outperforms the baselines in this attribute. On the other hand, the general capability of embedding is primarily established with **C-MTP (unlabeled)**, as BGE-*pretrain*'s performance is close to the baselines on the rest of the aspects, like STS and Clustering. *This puts our embedding models in a very favorable position for further improvements.*

As for **C-MTP (labeled)**, the dataset is much smaller but of better quality. With another round of fine-tuning on **C-MTP (labeled)**, the empirical advantage is significantly expanded for the final checkpoint BGE-*finetune*, where it gives rise to a jump in average performance from 59.0 (BGE-*pretrain*) to 63.96 (BGE-*finetune*). Knowing that the text pairs in **C-MTP (labeled)** are mainly gathered from retrieval and NLI tasks, the most notable improvements are achieved on closely related tasks, namely retrieval, re-ranking, STS, and pair classification. On other tasks, it preserves or marginally improves performance. *This indicates that a mixture of high-quality and diversified labeled data is able to bring forth substantial and comprehensive improvements for a pre-trained embedding model.*

Table 4: Impact of batch size.

Task \ Batch Size	256	2,048	19,200
Retrieval	57.25	60.96	**63.90**
STS	46.16	46.60	**47.71**
Pair CLF	**62.02**	61.91	61.67
CLF	65.71	67.42	**68.59**
Re-rank	58.59	59.98	**60.12**
Cluster	**49.52**	49.04	47.73
Average	56.43	57.92	**59.00**

We make further exploration about our **training recipe**, particularly the impact from contrastive learning, task-specific fine-tuning, and pre-training.

One notable feature of our training recipe is that we adopt a large batch size for contrastive learning. According to previous studies, the learning of the embedding model may benefit from the increasing of negative samples [22, 32, 43]. Given our dependency on in-batch negative samples, the batch size needs to be expanded as much as possible. In our implementation, we use a compound strategy of gradient checkpointing and cross-device embedding sharing [18], which results in a maximum batch size of 19,200. By making a parallel comparison between bz: 256, 2028, 19,200, we *observe consistent improvement in embedding quality with the expansion of batch size* (noted as bz). The most notable improvement is achieved in retrieval performance. This is likely due to the fact that retrieval is usually performed over a large database, where embeddings need to be highly discriminative.

Another feature is the utilization of instructions during task-specific fine-tuning. The task-specific instruction serves as a hard prompt. It differentiates the embedding model's activation, which lets the model better accommodate a variety of different tasks. We perform the ablation study by removing this operation, noted as "*w.o. Instruct*". Compared with this variation, the original method

Table 5: Performance of English Models on MTEB.

Model Name	Dim.	Average	Retrieval	Cluster	Pair CLF	Re-rank	STS	Summarize	CLF
GTE (large)	1024	63.13	52.22	**46.84**	85.00	59.13	83.35	31.66	73.33
GTE (base)	768	62.39	51.14	46.2	84.57	58.61	82.3	31.17	73.01
E5 (large)	1024	62.25	50.56	44.49	86.03	56.61	82.05	30.19	75.24
Instructor-XL	768	61.79	49.26	44.74	86.62	57.29	83.06	32.32	61.79
E5 (base)	768	61.5	50.29	43.80	85.73	55.91	81.05	30.28	73.84
GTE (small)	384	61.36	49.46	44.89	83.54	57.7	82.07	30.42	72.31
OpenAI Ada 002	1536	60.99	49.25	45.9	84.89	56.32	80.97	30.8	70.93
E5 (small)	384	59.93	49.04	39.92	84.67	54.32	80.39	31.16	72.94
ST5 (XXL)	768	59.51	42.24	43.72	85.06	56.42	82.63	30.08	73.42
MPNet (base)	768	57.78	43.81	43.69	83.04	59.36	80.28	27.49	65.07
SGPT Bloom (7.1B)	4096	57.59	48.22	38.93	81.9	55.65	77.74	**33.60**	66.19
BGE (small)	384	62.17	51.68	43.82	84.92	58.36	81.59	30.12	74.14
BGE (base)	768	63.55	53.25	45.77	86.55	58.86	82.4	31.07	75.53
BGE (large)	1024	**64.23**	**54.29**	46.08	**87.12**	**60.03**	**83.11**	31.61	**75.97**

BGE-f gives rise to better average performance. Besides, there are more significant empirical advantages on retrieval, STS, pair classification, and re-rank. All these perspectives are closely related to the training data at the final stage, i.e. **C-MTP (labeled)**, where the model is fine-tuned on a small group of tasks. This indicates that *using instructions may substantially contribute to the quality of task-specific fine-tuning.*

One more characteristic is that we use a specifically pre-trained text encoder to train **BGE**, rather than using common choices, like BERT [15] and RoBERTa [29]. To explore its impact, we replace the pre-trained text encoder with the widely used Chinese-RoBERTa[15], noted as "BGE w.o. pre-train". According to the comparison between BGE-*pretrain* and BGE w.o. pre-train, *the using of pre-trained text encoder notably improves the retrieval capability, while preserving similar performances on other perspectives.*

REFERENCES

[1] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*. 252–263.

[2] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M Cer, Mona T Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity.. In *SemEval@ COLING*. 81–91.

[3] Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics).

[4] Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. 385–393.

[5] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. * SEM 2013 shared task: Semantic textual similarity. In *Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*. 32–43.

[6] Loubna Ben Allal, Raymond Li, Denis Kocetkov, Chenghao Mou, Christopher Akiki, Carlos Munoz Ferrandis, Niklas Muennighoff, Mayank Mishra, Alex Gu, Manan Dey, et al. 2023. SantaCoder: don't reach for the stars! *arXiv preprint arXiv:2301.03988* (2023).

[7] Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. 2022. Task-aware retrieval with instructions. *arXiv preprint arXiv:2211.09260* (2022).

[8] 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* (2021).

[9] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*. PMLR, 2206–2240.

[10] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* (2015).

[11] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[12] 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* (2022).

[13] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416* (2022).

[14] Alexis Conneau and Douwe Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. *arXiv preprint arXiv:1803.05449* (2018).

[15] 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* (2018).

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

[17] Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. *A framework for few-shot language model evaluation.* https://doi.org/10.5281/zenodo.5371628

[18] Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021. Scaling deep contrastive learning batch size under memory limited setup. *arXiv preprint arXiv:2101.06983* (2021).

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

[20] Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, et al. 2017. Dureader: a chinese machine reading comprehension dataset from real-world applications. *arXiv preprint arXiv:1711.05073* (2017).

[21] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes

[15] huggingface.co/hfl/chinese-roberta-wwm-ext-large

Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556* (2022).

[22] 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* (2021).

[23] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299* (2022).

[24] Ehsan Kamalloo, Nandan Thakur, Carlos Lassance, Xueguang Ma, Jheng-Hong Yang, and Jimmy Lin. 2023. Resources for Brewing BEIR: Reproducible Reference Models and an Official Leaderboard. *arXiv preprint arXiv:2306.07471* (2023).

[25] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[26] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[27] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. 2023. StarCoder: may the source be with you! *arXiv preprint arXiv:2305.06161* (2023).

[28] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. *arXiv preprint arXiv:2308.03281* (2023).

[29] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

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

[31] Dingkun Long, Qiong Gao, Kuan Zou, Guangwei Xu, Pengjun Xie, Ruijie Guo, Jian Xu, Guanjun Jiang, Luxi Xing, and Ping Yang. 2022. Multi-cpr: A multi domain Chinese dataset for passage retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3046–3056.

[32] Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904* (2022).

[33] Niklas Muennighoff, Qian Liu, Armel Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro von Werra, and Shayne Longpre. 2023. OctoPack: Instruction Tuning Code Large Language Models. *arXiv preprint arXiv:2308.07124* (2023).

[34] Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. Scaling Data-Constrained Language Models. *arXiv preprint arXiv:2305.16264* (2023).

[35] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. MTEB: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316* (2022).

[36] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786* (2022).

[37] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005* (2022).

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

[39] Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877* (2021).

[40] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899* (2021).

[41] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. *arXiv preprint arXiv:2307.16789* (2023).

[42] 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* (2022).

[43] 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* (2020).

[44] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446* (2021).

[45] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[46] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[47] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207* (2021).

[48] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652* (2023).

[49] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615* (2022).

[50] Hongjin Su, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, Tao Yu, et al. 2022. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741* (2022).

[51] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663* (2021).

[52] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Simlm: Pre-training with representation bottleneck for dense passage retrieval. *arXiv preprint arXiv:2207.02578* (2022).

[53] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533* (2022).

[54] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021).

[55] Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426* (2017).

[56] Shitao Xiao, Zheng Liu, Weihao Han, Jianjin Zhang, Defu Lian, Yeyun Gong, Qi Chen, Fan Yang, Hao Sun, Yingxia Shao, et al. 2022. Distill-vq: Learning retrieval oriented vector quantization by distilling knowledge from dense embeddings. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1513–1523.

[57] Shitao Xiao, Zheng Liu, Weihao Han, Jianjin Zhang, Yingxia Shao, Defu Lian, Chaozhuo Li, Hao Sun, Denvy Deng, Liangjie Zhang, et al. 2022. Progressively optimized bi-granular document representation for scalable embedding based retrieval. In *Proceedings of the ACM Web Conference 2022*. 286–296.

[58] Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2023. RetroMAE-2: Duplex Masked Auto-Encoder For Pre-Training Retrieval-Oriented Language Models. *arXiv preprint arXiv:2305.02564* (2023).

[59] Shitao Xiao, Zheng Liu, Yingxia Shao, Defu Lian, and Xing Xie. 2021. Matching-oriented product quantization for ad-hoc retrieval. *arXiv preprint arXiv:2104.07858* (2021).

[60] 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* (2023).

[61] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[62] 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* (2020).

[63] 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 (2021), 65–68.

[64] Jianjin Zhang, Zheng Liu, Weihao Han, Shitao Xiao, Ruicheng Zheng, Yingxia Shao, Hao Sun, Hanqing Zhu, Premkumar Srinivasan, Weiwei Deng, et al. 2022. Uni-retriever: Towards learning the unified embedding based retriever in bing sponsored search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4493–4501.

[65] S. Zhang, X. Zhang, H. Wang, L. Guo, and S. Liu. 2018. Multi-Scale Attentive Interaction Networks for Chinese Medical Question Answer Selection. *IEEE Access* 6 (2018), 74061–74071. https://doi.org/10.1109/ACCESS.2018.2883637