C-Pack: Packaged Resources To Advance General Chinese Embedding

Anonymous ACL submission

Abstract

We introduce **C-Pack**, a package of resources that significantly advance the field of general Chinese embeddings. **C-Pack** includes three critical resources. 1) **C-MTEB** is a comprehensive benchmark for Chinese text embeddings covering 6 tasks and 35 datasets. 2) **C-MTP** is a massive text embedding dataset curated from labeled and unlabeled Chinese corpora for training embedding models. 3) **C-TEM** is a family of embedding models covering multiple sizes. Our models outperform all prior Chinese text embeddings on **C-MTEB** by up to +10% upon the time of the release. We also integrate and optimize the entire suite of training methods for **C-TEM**. Along with our resources on general Chinese embedding, we release our data and models for English text embeddings. The English models outperform all existing embedding models on the MTEB benchmark; meanwhile, our released English data is 2 times larger than the Chinese data. All these resources will be made publicly available.

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 (Karpukhin et al., 2020; Lewis et al., 2020; Guu et al., 2020). 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 (Borgeaud et al., 2022; Qin et al., 2023).

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:

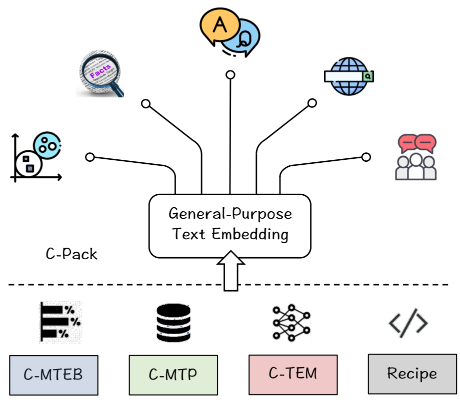

Figure 1: **The C-Pack resources to support general Chinese embedding.**

• **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 (Izacard et al., 2021; Ni et al., 2021b; Wang et al., 2022b), which is orders of magnitude greater than typical task-specific datasets, like MS MARCO (Nguyen et al., 2016) and NLI (Bowman et al., 2015; Williams et al., 2017). Besides scale, the training data needs to be collected from a wide range of sources so as to improve the generality across different tasks (Izacard et al., 2021; Wang et al., 2022b). 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 (Wang et al., 2022b).

• **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 (Devlin et al., 2018) and T5 (Raffel et al., 2020), the quality of text embedding can be substantially improved by pre-training with large-scale unlabeled data (Izacard et al., 2021; Wang et al., 2022b). 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 (Gao and Callan, 2021), contrastive learning with sophisticated negative sampling to improve the embedding's discriminability (Qu et al., 2020), and instruction-based fine-tuning (Su et al., 2022; Asai et al., 2022) 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 (Thakur et al., 2021) 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 (Muennighoff et al., 2022a) 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 progress has been achieved in this field, such as work from Contriever (Izacard et al., 2021), E5 (Wang et al., 2022b), and OpenAI Text Embedding (Neelakantan et al., 2022). However, most of these works are oriented to the English world. In contrast, there is a severe shortage of competitive models for general Chinese embedding due to a series of limitations: there are neither well-prepared training resources nor suitable benchmarks to evaluate the generality.

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.[1] **C-MTEB** collects 35 public-available datasets belonging to 6 types of tasks. 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, which integrates both labeled data and unlabeled data curated from Wudao (Yuan et al., 2021), one of the largest corpora for pre-training Chinese language models. **C-MTP** is not only large and diverse but also cleaned to ensure the data quality.

• **C-TEM** (Chinese Text Embedding Models). 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: **C-TEM** outperforms all previously Chinese text embedding models on all aspects of **C-MTEB** by large margins. Besides being directly applicable, **C-TEM** can also be fine-tuned with additional data for better task-specific performances.

• **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.

In summary, C-Pack provides a go-to option for people's **application** of general-purpose Chinese text embedding. It substantially advances the **training** and **evaluation**, laying a solid foundation for the future development of this field.

2 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 **C-TEM**. 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.

2.1 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

1. https://huggingface.co/spaces/mteb/leaderboard

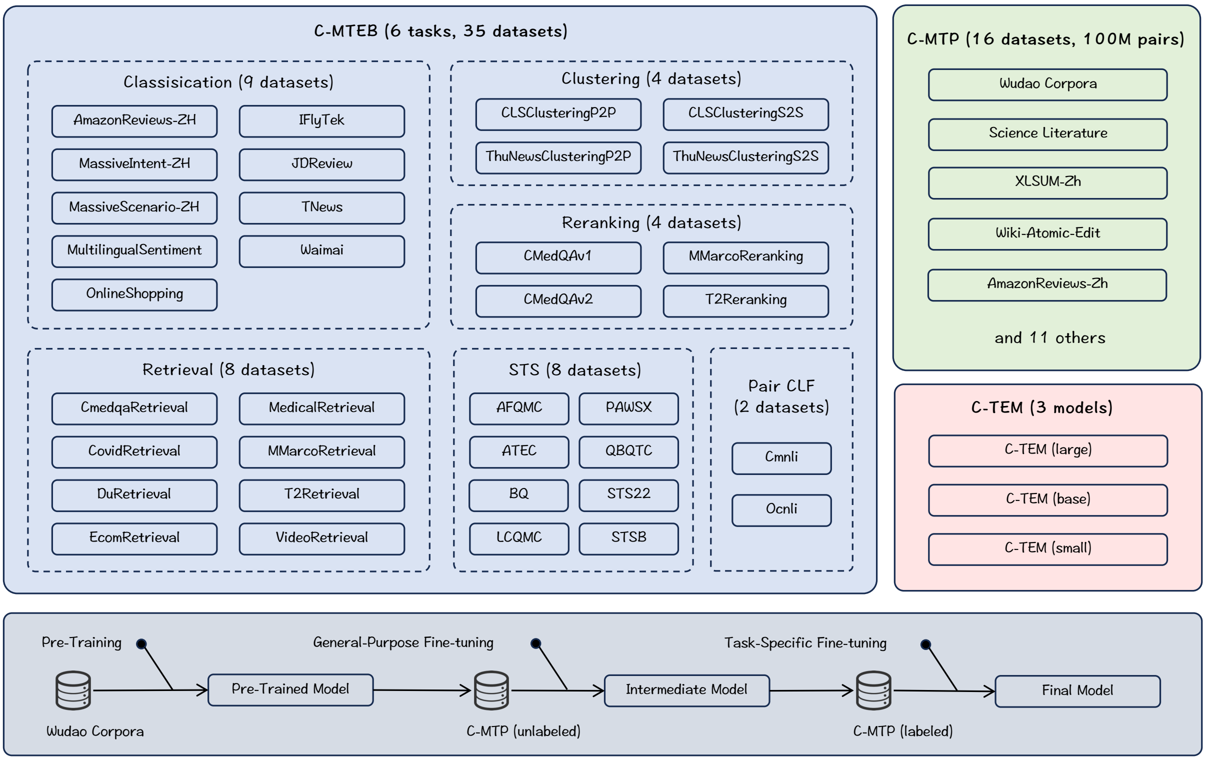

Figure 2: **Overview of C-Pack. C-MTEB** is a benchmark for Chinese text embeddings. **C-MTP** is a large-scale Chinese embedding training dataset. **C-TEM** are state-of-the-art Chinese embedding models. The training recipe is shown at the bottom.

has put forward essential datasets to study Chinese text embeddings, such as CMNLI (Xu et al., 2020a), DuReader (He et al., 2017), T^2Ranking (Xie et al., 2023). However, these datasets are independently curated and only focus on one specific capability of the text embeddings. Thus, we create **C-MTEB** to **1)** comprehensive collect related datasets, **2)** categorize the datasets and **3)** standardize and integrate the evaluation pipleines.

In particular, we collect a total of 35 public datasets, all of which can be used to evaluate Chinese text embeddings. 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 task and its 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 (Thakur et al., 2021), using NDCG@10 as the main metric.

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

- **STS** (Semantic Textual Similarity). The STS (Agirre et al., 2012, 2013, 2014, 2015, 2016) task is to measure the correlation of two sentences based on their embedding similarity. Following the original setting in Sentence-BERT (Reimers and Gurevych, 2019), the Spearman's correlation is computed with the given label, whose result is used as the main metric.

- **Classification**. The classification task reuses the logistic regression classifier from MTEB (Muennighoff et al., 2022a), where the provided label is predicted based on the input embedding.

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

Table 1: **Composition of C-MTP**

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 (Muennighoff et al., 2022a), 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**.

2.2 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 curation of massive unlabeled data, *a.k.a.* **C-MTP (unlabeled)**; and the comprehensive collection of labeled data, *a.k.a.* **C-MTP (labeled)**. 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 Wudao corpus (Yuan et al., 2021), which is the largest well-formatted dataset for pre-training Chinese language models. For each of its articles, we extract (title, passage) to form a text pair. Following the same recipe, we also collect such text pairs from other similar web content like Zhihu, Baike, news websites, etc. Aside from the open web content, we also explore other public Chinese datasets to extract text pairs, such as CSL (scientific literature), Amazon-Review-Zh (reviews), Wiki Atomic Edits (paraphrases), CMRC (machine reading comprehension), XLSUM-Zh (summarization), etc. The paired structures are obvious in these datasets, which are directly extracted for the augmentation of **C-MTP (unlabeled)**.

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 use a simple strategy to filter the data before adding it to **C-MTP (unlabeled)**. Particularly, we use a third-party model: Text2Vec-Chinese[2] 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 following labeled datasets are collected for **C-MTP (labeled)** due to their quality and diversity: T^2-Ranking (Xie et al., 2023), DuReader (He et al., 2017; Qiu et al., 2022), mMARCO (Bonifacio et al., 2021), and NLI-Zh[3] (which includes ATEC[4], BQ[5], LCQMC[6], PAWSX[7], CNSD[8]). There are 838,465 paired texts in total. 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. Besides, **C-MTP (labeled)** also fully covers different capabilities of the text embedding, like retrieval, ranking, similarity comparison, etc., which helps to improve the embedding model's generality after fine-tuning.

Given the differences in scale and quality, **C-MTP (unlabeled)** and **C-MTP (labeled)** are applied to different training stages, which jointly re-

2. https://huggingface.co/GanymedeNil
3. https://huggingface.co/datasets/shibing624/nli_zh
4. https://github.com/IceFlameWorm/NLP_Datasets/tree/master/ATEC
5. http://icrc.hitsz.edu.cn/info/1037/1162.htm
6. http://icrc.hitsz.edu.cn/info/1037/1162.htm
7. https://arxiv.org/abs/1908.11828
8. https://github.com/pluto-junzeng/CNSD

sult in a strong performance for the embedding model. Detailed analysis will be made in our training recipe.

2.3 Model Class: C-TEM

We provide a comprehensive class of well-trained embedding models for the community. Our models take a BERT-like architecture, where the last layer's hidden state of the special token [CLS] is trained to work as the embedding. There are three different scales for the models: 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 C-TEM; 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 C-TEM have been well-trained and achieve a strong generality for a wide variety of tasks. Meanwhile, they can also be further fine-tuned if 1) the embeddings are applied for a specific scenario, 2) the training data is presented for the application scenario. 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 C-TEM, and the fine-tuned models from other general pre-trained encoders, like BERT. In other words, C-TEM not only presents people with direct usage embeddings but also works as a foundation where people may develop more powerful embeddings.

2.4 Training Recipe

The training recipe of C-TEM 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)**, whose specifications are made as follows.

• **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 (Yuan et al., 2021), which is a huge and high-quality dataset for Chinese language model pre-training. We leverage the MAE-style approach presented in RetroMAE (Liu and Shao, 2022; Xiao et al., 2023), 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:

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

(Enc, Dec are the encoder and decoder, 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 (Karpukhin et al., 2020) 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 (Su et al., 2022; Asai et al., 2022), 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 (Xiong et al., 2020).

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
TEM (small)	512	63.07	49.45	70.35	63.64	61.48	45.09	58.28
TEM (base)	768	69.53	54.12	77.50	67.07	64.91	47.63	62.80
TEM (large)	1024	**71.53**	**54.98**	**78.94**	**68.32**	**65.11**	48.39	**63.96**

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

3 Experiments

We conduct experimental studies for the following purposes. **P1**. The extensive evaluation of different Chinese text embeddings on **C-MTEB**. **P2**. The empirical verification of the text embeddings by **C-TEM**. **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[9] base and large; Luotuo[10]; M3E[11] base and large; multilingual E5 (Wang et al., 2022b) and OpenAI text embedding ada 002[12]. The main metric presented in Section 2.1 is reported for each task in **C-MTEB**.

3.1 General Evaluation

We extensively evaluate **C-TEM** against popular Chinese text embeddings on **C-MTEB** as shown in Table 2.[13] We 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 **C-TEM**. *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 high-precision usages, or switch to the small-scale one for high-throughput scenarios.

3.2 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.

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 intermedi-

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

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
w.o. Instruct	1024	70.55	53.00	76.77	68.58	64.91	**50.01**	63.40
TEM-i	1024	63.90	47.71	61.67	**68.59**	60.12	47.73	59.00
TEM-i w.o. pre-train	1024	62.56	48.06	61.66	67.89	61.25	46.82	58.62
TEM-f	1024	**71.53**	**54.98**	**78.94**	68.32	**65.11**	48.39	**63.96**

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

ate checkpoint, denoted as **TEM-i**. 2) **C-MTP (labeled)**, where the task-specific fine-tuning is further conducted on top of TEM-i; the model produced from this stage is called the final checkpoint, noted as **TEM-f**. 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, TEM-i 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 TEM-i notably outperforms the baselines in this attribute. On the other hand, the general capability of embedding is primarily established with **C-MTP (unlabeled)**, as TEM-i'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 TEM-f, where it gives rise to a jump in average performance from 59.0 (TEM-i) to 63.96 (TEM-f). 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.*

We further explore the impact of our **training recipe**, particularly 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 (Izacard et al., 2021; Qu et al., 2020; Muennighoff, 2022). 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 (Gao et al., 2021b), 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 TEM-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 task-specific fine-tuning.*

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**

Table 4: **Impact of batch size.**

One more characteristic is that we use a specifically pre-trained text encoder to train **C-TEM**, rather than using common choices, like BERT and RoBERTa (Liu et al., 2019). To explore its impact, we replace the pre-trained text encoder with the widely used Chinese-RoBERTa[14], noted as "TEM-i w.o. pre-train". According to the comparison with TEM-i, the *pre-trained text encoder notably improves the retrieval capability, while preserving similar performances on other aspects.*

4 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 (Karpukhin et al., 2020) but also due to its fundamental role in augmenting large language models (Lewis et al., 2020; Guu et al., 2020; Borgeaud et al., 2022; Izacard et al., 2022; Shi et al., 2023). 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 (Izacard et al., 2021), GTR (Ni et al., 2021b), sentence-T5 (Ni et al., 2021a), Sentence-Transformer (Reimers and Gurevych, 2019), E5 (Wang et al., 2022a), OpenAI text embedding (Neelakantan et al., 2022), 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 (Izacard et al., 2021; Wang et al., 2022b; Neelakantan et al., 2022). Secondly, the embedding model must be scaled up, as large text encoders are more generalizable across different application scenarios (Muennighoff, 2022; Ni et al., 2021b,a) in line with observations for the importance of scaling LLMs (Hoffmann et al., 2022; Rae et al., 2021; Brown et al., 2020; Chowdhery et al., 2022; Srivastava et al., 2022; Gao et al., 2021a; Li et al., 2023a; Allal et al., 2023; Muennighoff et al., 2023b). Thirdly, the training recipe must be optimized through pre-training (Liu and Shao, 2022; Wang et al., 2022a), negative sampling (Izacard et al., 2021; Wang et al., 2022a), and multi-task fine-tuning (Su et al., 2022; Asai et al., 2022; Sanh et al., 2021; Wei et al., 2021; Muennighoff et al., 2022b, 2023a; Chung et al., 2022). Aside from the above, it is also critical to establish proper benchmarks to evaluate the generality of text embeddings. Unlike previous task-specific evaluations, like MS-MARCO (Nguyen et al., 2016), SentEval (Conneau and Kiela, 2018), 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 (Thakur et al., 2021; Kamalloo et al., 2023), where the embeddings can be evaluated across different retrieval tasks. It is later extended by MTEB (Muennighoff et al., 2022a), where all major aspects of text embeddings can be comprehensively evaluated.

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 is crucially important.

5 Conclusion

We present C-Pack to advance progress towards general Chinese embedding. C-Pack consists of three core resources **1)** The benchmark **C-MTEB**, covering 6 major tasks of embeddings and 35 datasets, making it the most comprehensive benchmark to evaluate the generality of Chinese embeddings. **2)** The training data **C-MTP**, curated from massive unlabeled corpora and high-quality labeled datasets. Its unprecedented scale, diversity, and quality contribute to the superior generality of our embedding models. **3)** The models **C-TEM**, which are empirically competitive. Their different sizes provide people with the flexibility to trade off efficiency and embedding quality. The entire training recipe is also provided along with these resources. The public release of C-Pack facilitates the usage of general Chinese embedding and also paves the way for its future advancement.

14. huggingface.co/hfl/chinese-roberta-wwm-ext-large

6 Limitations and Risks

In future work, our study can be enhanced from the following perspectives. 1) Improvement of data quality, possibly with the introduction of more data cleaning heuristics and model-based methods. 2) Expansion of dataset, by collecting training data from more diversified domains and even other languages. 3) Exploring and developing models with higher generality, e.g., embeddings which can support all languages and data modalities. Given the dependency on public datasets, like Wudao (Yuan et al., 2021) and C4 (Raffel et al., 2020), our resource is likely to exhibit similar ethical risks, including social biases and toxic statements, which should be addressed in future research.

References

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)*, pages 252–263.

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*, pages 81–91.

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).

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)*, pages 385–393.

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*, pages 32–43.

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*.

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*.

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*.

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*, pages 2206–2240. PMLR.

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*.

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*, 33:1877–1901.

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

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 2018. The bq corpus: A large-scale domain-specific chinese corpus for sentence semantic equivalence identification. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 4946–4951.

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*.

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*.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*.

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*.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, et al. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. *arXiv preprint arXiv:2204.08582*.

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. 2021a. A framework for few-shot language model evaluation.

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

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

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

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.

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*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

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*.

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*.

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*.

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*.

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*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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*, 33:9459–9474.

Jingyang Li and Maosong Sun. 2007. Scalable term selection for text categorization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 774–782.

Jingyang Li, Maosong Sun, and Xian Zhang. 2006. A comparison and semi-quantitative analysis of words and character-bigrams as features in chinese text categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 545–552.

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

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*.

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

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.

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*.

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

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*, pages 3046–3056.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. RecSys '13, New York, NY, USA. Association for Computing Machinery.

Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

Niklas Muennighoff, Qian Liu, Armel Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro von Werra, and Shayne Longpre. 2023a. Octopack: Instruction tuning code large language models. *arXiv preprint arXiv:2308.07124*.

Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023b. Scaling data-constrained language models. *arXiv preprint arXiv:2305.16264*.

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

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. 2022b. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

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*.

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.

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

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. 2021b. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

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*.

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*.

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*.

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):5485–5551.

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

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*.

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

Teven Le Scao, Thomas Wang, Daniel Hesslow, Lucile Saulnier, Stas Bekman, M Saiful Bari, Stella

Bideman, Hady Elsahar, Niklas Muennighoff, Jason Phang, et al. 2022b. What language model to train if you have one million gpu hours? *arXiv preprint arXiv:2210.15424*.

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*.

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*.

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*.

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*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

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

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

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*.

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

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*.

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*.

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*.

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

Liang Xu, Xuanwei Zhang, and Qianqian Dong. 2020b. Cluecorpus2020: A large-scale chinese corpus for pre-training language model. *arXiv preprint arXiv:2003.01355*.

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*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

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.

Sheng Zhang, Xin Zhang, Hui Wang, Jiajun Cheng, Pei Li, and Zhaoyun Ding. 2017. Chinese medical question answer matching using end-to-end character-level multi-scale cnns. *Applied Sciences*, 7(8):767.

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

A C-MTEB Datasets

Name	URL	Description	Categ.	Test Samples
		Classification		
AmazonReviewsClassification (Muennighoff et al., 2022a; McAuley and Leskovec, 2013)	https://hf.co/datasets/mteb/amazon_reviews_multi	Sentiment of Amazon reviews	s2s	5,000
IFlyTek (Xu et al., 2020a)	https://hf.co/datasets/anonymous/IFlyTek-classification	Long text classification for App descriptions	s2s	2,600
JDReview (https://hf.co/datasets/kuroneko5943/jd21)	https://hf.co/datasets/anonymous/JDReview-classification	iPhone reviews	s2s	533
MassiveIntentClassification (Muennighoff et al., 2022a; FitzGerald et al., 2022)	https://hf.co/datasets/mteb/amazon_massive_intent	Amazon Alexa virtual assistant utterances annotated with the associated intent	s2s	16,500
MassiveScenarioClassification (Muennighoff et al., 2022a; FitzGerald et al., 2022)	https://hf.co/datasets/mteb/amazon_massive_scenario	Amazon Alexa virtual assistant utterances annotated with the associated scenario	s2s	16,500
MultilingualSentiment (McAuley and Leskovec, 2013)	https://hf.co/datasets/anonymous/MultilingualSentiment-classification	Sentiment of Amazon reviews	s2s	3,000
OnlineShopping (https://github.com/SophonPlus/ChineseNlpCorpus/blob/master/datasets/online_shopping_10_cats/intro.ipynb)	https://hf.co/datasets/anonymous/OnlineShopping-classification	Sentiment Analysis of User Reviews on Online Shopping Websites	s2s	1,000
TNews (Xu et al., 2020a)	https://hf.co/datasets/anonymous/TNews-classification	Short Text Classification for News	s2s	10,000
Waimai (https://github.com/SophonPlus/ChineseNlpCorpus/blob/master/datasets/waimai_10k/intro.ipynb)	https://hf.co/datasets/anonymous/waimai-classification	Sentiment Analysis of user reviews on takeaway platforms	s2s	1,000
		Clustering		

CLSClusteringP2P (Li et al., 2022)	https://hf.co/datasets/anonymous/CLSClusteringP2P	Clustering of titles + abstract from CLS dataset. Clustering of 13 sets, based on the main category.	p2p	10,000
CLSClusteringS2S (Li et al., 2022)	https://hf.co/datasets/anonymous/anonymous/CLSClusteringS2S	Clustering of titles from CLS dataset. Clustering of 13 sets, based on the main category.	s2s	10,000
ThuNewsClusteringP2P (Li et al., 2006; Li and Sun, 2007)	https://hf.co/datasets/anonymous/ThuNewsClusteringP2P	Clustering of titles + abstract from the THUCNews dataset	p2p	10,000
ThuNewsClusteringS2S (Li et al., 2006; Li and Sun, 2007)	https://hf.co/datasets/anonymous/ThuNewsClusteringS2S	Clustering of titles from the THUCNews dataset	s2s	10,000
Pair Classification				
Cmnli (Xu et al., 2020a,b; Conneau and Kiela, 2018; Williams et al., 2017)	https://hf.co/datasets/anonymous/CMNLI	Chinese Multi-Genre NLI	s2s	139,000
Ocnli (Hu et al., 2020)	https://hf.co/datasets/anonymous/OCNLI	Original Chinese Natural Language Inference dataset	s2s	3,000
Reranking				
T2Reranking (Xie et al., 2023)	https://hf.co/datasets/anonymous/T2Reranking	T2Ranking: A large-scale Chinese Benchmark for Passage Ranking	s2p	24,382
MMarcoRetrieval (Bonifacio et al., 2021)	https://hf.co/datasets/anonymous/Mmarco-reranking	mMARCO is a multilingual version of the MS MARCO passage ranking dataset	s2p	7,437
CMedQAv1 (Zhang et al., 2017)	https://hf.co/datasets/anonymous/CMedQAv1-reranking	Chinese community medical question answering	s2p	2,000
CMedQAv2 (Zhang et al., 2018)	https://hf.co/datasets/anonymous/anonymous/CMedQAv2-reranking	Chinese community medical question answering	s2p	4,000
Retrieval				

T2Retrieval (Xie et al., 2023)	https://hf.co/datasets/anonymous/T2Retrieval	T2Ranking: A large-scale Chinese Benchmark for Passage Ranking	s2p	24,832
MMarcoRetrieval (Bonifacio et al., 2021)	https://hf.co/datasets/anonymous/MMarcoRetrieval	mMARCO is a multilingual version of the MS MARCO passage ranking dataset	s2p	7,437
DuRetrieval (Qiu et al., 2022)	https://hf.co/datasets/anonymous/DuRetrieval	A Large-scale Chinese Benchmark for Passage Retrieval from Web Search Engine	s2p	4,000
CovidRetrieval (Qiu et al., 2022)	https://hf.co/datasets/anonymous/CovidRetrieval	COVID-19 news articles	s2p	949
CmedqaRetrieval (Qiu et al., 2022)	https://hf.co/datasets/anonymous/CmedqaRetrieval	Online medical consultation text	s2p	3,999
EcomRetrieval (Long et al., 2022)	https://hf.co/datasets/anonymous/EcomRetrieval	Passage retrieval dataset collected from Alibaba search engine systems in e-commerce domain	s2p	1,000
MedicalRetrieval (Long et al., 2022)	https://hf.co/datasets/anonymous/MedicalRetrieval	Passage retrieval dataset collected from Alibaba search engine systems in medical domain	s2p	1,000
VideoRetrieval (Long et al., 2022)	https://hf.co/datasets/anonymous/VideoRetrieval	Passage retrieval dataset collected from Alibaba search engine systems in video domain	s2p	1,000
STS				
AFQMC (Xu et al., 2020a)	https://hf.co/datasets/anonymous/AFQMC	Ant Financial Question Matching Corpus	s2s	3,861
ATEC (https://github.com/IceFlameWorm/NLP_Datasets/tree/master/ATEC)	https://hf.co/datasets/anonymous/ATEC	ATEC NLP sentence pair similarity competition	s2s	20,000
BQ (Chen et al., 2018)	https://hf.co/datasets/anonymous/BQ	Bank Question Semantic Similarity	s2s	10,000

LCQMC (Liu et al., 2018)	https://hf.co/datasets/anonymous/LCQMC	A large-scale Chinese question matching corpus.	s2s	12,500
PAWSX (Yang et al., 2019)	https://hf.co/datasets/anonymous/PAWSX	Translated PAWS evaluation pairs	s2s	2,000
QBQTC (Xu et al., 2020a)	https://hf.co/datasets/anonymous/QBQTC	QQ Browser Query Title Corpus	s2s	5,000
STSB (Cer et al., 2017)	https://hf.co/datasets/anonymous/STSB	Translate STS-B into Chinese	s2s	1,360
STS-22 (Muennighoff et al., 2022a)	https://hf.co/datasets/mteb/sts22-crosslingual-sts	Chinese news	p2p	656

Table 5: **Overview of datasets in C-MTEB.**

B C-MTP Composition

We mine large-scale pairs of data from various domains. Table 6 shows the details for each data.

data source	type of text pairs	# of pairs	URL
cmrc2018	(query, context)	9,669	https://huggingface.co/datasets/cmrc2018
dureader	(query, context)	96,486	https://github.com/baidu/DuReader
simclue	(sentence$_a$, sentence$_b$)	388,779	https://github.com/CLUEbenchmark/SimCLUE
csl	(title, abstract)	394,846	https://arxiv.org/abs/2209.05034
amazon_reviews_multi	(title, body)	157,762	https://huggingface.co/datasets/amazon_reviews_multi
wiki_atomic_edits	(sentence, edited sentence)	1,213,688	https://huggingface.co/datasets/wiki_atomic_edits
mlqa	(question, context)	70,594	https://huggingface.co/datasets/mlqa
xlsum	(title, summary) (title, text)	89,505	https://huggingface.co/datasets/csebuetnlp/xlsum
wudao	(title, passage)	37,318,330	https://data.baai.ac.cn/details/WuDaoCorporaText
Misc	–	60,260,341	–

Table 6: **Details for each dataset.** The Misc data comes from the Internet, including QA, paper, and news data.

C English Models

Using our recipe, we also train a set of English text embedding models presented in Table 7. At the time of writing, our English TEM models are state-of-the-art on the English MTEB benchmark (Muennighoff et al., 2022a) across its 56 datasets. Our models outperform significantly larger models, such as SGPT Bloom which has 7.1 billion parameters (Muennighoff, 2022; Scao et al., 2022a,b). We advance the prior state-of-the-art by an absolute 1.1 (Li et al., 2023b). Our training recipe is the same as for our Chinese models, except for the usage of English data. We first finetune on unsupervised datasets including datasets like Wikipedia, CC-net, StackExchange, Reddit, S2orc, and datasets from sentence-transformers.[15] We then further fine-tune on supervised datasets including NLI (Gao et al., 2021c), FEVER (Thorne et al., 2018), NQ (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), Quora, StackExchange Duplicates and MEDI (Su et al., 2022).

15. https://huggingface.co/datasets/sentence-transformers/embedding-training-data

Model Name	Dim.	Average	Retrieval	Cluster	Pair CLF	Re-rank	STS	Summarize	CLF
TEM (large)	1024	**64.23**	**54.29**	46.08	**87.12**	**60.03**	**83.11**	31.61	**75.97**
TEM (base)	768	63.55	53.25	45.77	86.55	58.86	82.4	31.07	75.53
TEM (small)	384	62.17	51.68	43.82	84.92	58.36	81.59	30.12	74.14
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

Table 7: **Performance of English Models on MTEB.**