FPM: A Collection of Large-scale Foundation Pre-trained Language Models

Dezhou Shen (shendezhou@rct.ai)
rct ai https://orcid.org/0000-0001-5514-507X

Article

Keywords: language modelling, large scale modelling, language processing

Posted Date: November 8th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1061146/v1

FPM: A Collection of Large-scale Foundation Pre-trained Language Models

Dezhou Shen
Department of Computer Science
rct ai
Beijing, CN 101300
shendezhou@rct.ai

Abstract

Recent work in language modeling has shown that training large-scale Transformer models has promoted the latest developments in natural language processing applications. However, there is very little work to unify the current effective models. In this work, we use the current effective model structure to launch a model set through the current most mainstream technology. We think this will become the basic model in the future. For Chinese, using the GPT-2[9] model, a 10.3 billion parameter language model was trained on the Chinese dataset, and, in particular, a 2.9 billion parameter language model based on dialogue data was trained; the BERT model was trained on the Chinese dataset with 495 million parameters; the Transformer model has trained a language model with 5.6 billion parameters on the Chinese dataset. In English, corresponding training work has also been done. Using the GPT-2 model, a language model with 6.4 billion parameters was trained on the English dataset; the BERT[3] model trained a language model with 1.24 billion parameters on the English dataset, and in particular, it trained a 688 million parameter based on single card training technology Language model; Transformer model trained a language model with 5.6 billion parameters on the English dataset. In the TNEWS classification task evaluated by CLUE[13], the BERT-C model exceeded the 59.46% accuracy of ALBERT-xxlarge with an accuracy rate of 59.99%, an increase of 0.53%. In the QQP classification task evaluated by GLUE[11], the accuracy rate of 78.95% surpassed the accuracy rate of BERT-Large of 72.1%, an increase of 6.85%. Compared with the current accuracy rate of ERNIE, the first place in the GLUE evaluation of 75.2%, an increase of 3.75%.

1. Introduction

Natural language processing is developing rapidly, partly because of the increase in available calculations and the size of datasets. Abundant calculations and data make it possible to train larger and larger language models through unsupervised pre-training. As these models become larger, they exceed the memory limits of modern processors. Fortunately, it is possible to train a language model with hundreds of millions or even tens of billions of parameters on the NVIDIA A100 40GB GPU. By partitioning the model so that the weights and their associated optimizer states do not need to reside on the processor at the same time, several model parallelism methods overcome this limitation. By deploying multiple GPUs on one machine, we can train larger and more effective models. And now there are many methods to overcome model degradation, such as the technology of overcoming model degradation caused by model size scaling by rearranging the layer normalization and residual connection in the Transformer layer.

In summary, the contributions of this article are as follows:

For Chinese, using the GPT-2 model, a 10.3 billion parameter language model was trained on the Chinese dataset, which is the largest known Chinese generative model. And, in particular, trained a 2.9 billion parameter language model based on dialogue data. The BERT model has trained a 495 million parameter language model on the Chinese dataset and is the largest known Chinese coding model. The Transformer model trained a 5.6 billion parameter language model on the Chinese dataset.

In English, using the GPT-2 model, a language model with 6.4 billion parameters was trained on the English dataset; the BERT model trained a language model with 1.24 billion parameters on the English dataset. In particular, a language model with 688 million parameters trained based on single-card training technology is the largest known English coding model. The Transformer model trained a 2.9 billion parameter language model on the English dataset.

In the TNEWS classification task evaluated by CLUE, the BERT-C model exceeded the 59.46% accuracy of ALBERT-xxlarge[6] with an accuracy rate of 59.99%, an increase of 0.53%.

In the QQP classification task evaluated by GLUE, the accuracy rate of 78.95% surpassed the accuracy rate of BERT-Large of 72.1%, an increase of 6.85%. Compared with the current accuracy rate of ERNIE, the first place in the GLUE evaluation of 75.2%, an increase of 3.75%.

2. Related Work

2.1. Pre-trained Models Review

In terms of generative models, Tsinghua trained a 2.6 billion parameter CPM[17] model; Eleuther trained a 2.7 billion parameter GPT-Neo and a 6 billion parameter GPT-J[12] model; OpenAI trained a 175 billion parameter GPT-3[1] model.

In terms of coding models, iFLYTEK has trained a 330 million parameter Roberta[2] model, Alibaba trained a 27 billion parameter PLUG[7] model, Huawei trained a 110 billion parameter Pangu[15] model, and Saleforce trained a 1.6 billion parameter CTRL[5] model.

In terms of coding and decoding models, Tsinghua has trained a CPM2[16] model with 11 billion parameters.

3. GPT

In the experiment of this article, a multi-layer transformer decoder is used as the language model, which is a variant of the transformer. The model applies a multi-head self-attention operation to the input context label, and then a position feedforward layer to generate the output distribution on the target label. The decoder also consists of a stack of N=6 identical layers. In addition to the two sublayers in each encoder layer, the decoder also inserts a third sublayer that performs multi-head attention on the output of the encoder stack. Similar to the encoder, residual connections are used around each sub-layer, and then layer normalization is performed. The self-attention sublayer in the decoder stack is also modified to prevent the position from paying attention to subsequent positions. This masking, combined with the fact that the output embedding is offset by one position, ensures that the prediction of the position only depends on the known output at positions less than one.

3.1. Applications of Attention

Transformer uses multi-head attention in three different ways:

1. In the "encoder-decoder attention" layer, the query comes from the previous decoder layer, and the memory keys and values come from the encoder's output. This allows every position in the decoder to participate in all positions in the input sequence. This mimics the typical encoder-decoder attention mechanism in the sequence-to-sequence model.

2. The encoder includes a self-attention layer. In the self-attention layer, all keys, values, and queries come from the same place, in this case, the output of the previous layer in the encoder. Every position in the encoder can focus on all positions in the previous layer of the encoder.

3. Similarly, the self-attention layer in the decoder allows each position in the decoder to focus on all positions in the decoder up to and including that position. It is necessary to prevent leftward information flow in the decoder to preserve the autoregressive characteristics. This is achieved by masking all values corresponding to illegal connections in the softmax input.

4. BERT

BERT is a pre-trained Transformer network that sets up the latest results for various NLP tasks, including question answering, sentence classification, and sentence pair regression. The BERT input used for sentence pair regression consists of two sentences separated by a special [SEP] token. Apply more than 12 layers (base model) or 24 layers (large model) of multi-head attention, and pass the output to a simple regression function to derive the final label.

RoBERTa[8] shows that the performance of BERT can be further improved through small adjustments to the pre-training process. BERT and its detailed implementation will be introduced in this section. The framework has two steps: pre-training and fine-tuning. During pre-training, the model is trained on unlabeled data on different pre-training tasks. For fine-tuning, the BERT model is first initialized with pre-trained parameters, and then all parameters are fine-tuned using labeled data from downstream tasks. Each downstream task has a separate fine-tuning model, even if they are initialized with the same pre-training parameters.

5. Transformer

CPM-2 is a standard Transformer-based model that combines a two-way encoder and a one-way decoder. In order to effectively store model parameters on the GPU, model parallelism is used, which separates the self-attention layer and the feedforward layer along the width dimension, and finally distributes the partitions of a model on multiple GPUs. In order to reduce memory requirements and accelerate pre-training, mixed-precision training, gradient checkpointing, and zero stage 1 optimization are used.

Most competitive neural sequence transduction models have an encoder-decoder structure. Here, the encoder maps the input sequence of symbolic representation (x1,...,xn) to the continuous representation sequence z=(z1,...,zn). Given z, the decoder then generates the symbols of an output sequence (y1,...,ym), one element at a time. At each step, the model is automatically regressed, using previously

generated symbols as additional input when generating the next one. Transformer follows this overall architecture, using stacked self-attention and point-by-point, fully connected encoder and decoder layers, respectively.

Encoder: The encoder consists of a stack of N identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feedforward network. A residual connection is used around each of the two sublayers, followed by layer normalization. That is, the output of each sublayer is LayerNorm(x+Sublayer(x)), where Sublayer(x) is a function implemented by the sublayer itself. In order to facilitate these residual connections, all sub-layers and embedding layers in the model have produced an output of dimensional model.

Decoder: The decoder also consists of a stack of N identical layers. In addition to the two sublayers in each encoder layer, the decoder also inserts a third sublayer that performs multi-head attention on the output of the encoder stack. Similar to the encoder, residual connections are used around each sub-layer, and then layer normalization is performed. The self-attention sublayer in the decoder stack has also been modified to prevent locations from paying attention to subsequent locations. This masking, combined with the fact that the output embedding is offset by one position, ensures that the prediction of position only depends on the known output at positions less than 1.

6. Setup

The pre-trained language understanding model is the core task of natural language processing and language understanding. There are several forms of language modeling. In this work, we focus on GPT-2, a language model based on a left-to-right generative transformer; BERT, a bidirectional transformer model based on language model shielding; and CPM-2, a fusion editor Decoded transformer language model. The configuration of these models is explained in the next section, and refer to the original paper for more details.

6.1. Training Dataset

This work only focuses on model training, and uses the existing large-scale diversified dataset English dataset Pile[4] and Chinese dataset Wudao[14] on the dataset.

The GPT-English word segmenter uses the GPT-2 English vocabulary, which contains 30,000 symbols. It is trained by the OpenAI team using the 40GB text and 8 million documents collected by the WebText corpus. WebText is a web page curated and filtered by OpenAI humans. All outbound links with a rating of at least 3karma are crawled from Reddit. The generated dataset WebText contains a text subset of 45 million links. In the cleanup phase, links created after December 2017 were removed, and after deduplication and some heuristic-based cleanup, slightly more than 8 million documents were left, a total of 40GB of text, and all Wikipedia documents were deleted.

The GPT-Chinese word segmenter uses CPM's Chinese vocabulary, which contains 30,000 symbols. It is trained by the Tsinghua team using 100G multi-category texts, including encyclopedias, news, novels and questions and answers. The CPM Chinese vocabulary uses the unigram language model to build a new sub-word vocabulary based on the sub-word corpus, and since the length of the input sequence is usually greater than the length of a single document, different documents are connected by adding the "end-of-document" symbol after each document. Together to make full use of the input length. In the vocabulary construction process, a new sub-word vocabulary is constructed, including commonly used words and characters. Considering that the original BERT word segmentation will introduce an additional splitter between words, the CPM team set up a special token as a splitter to make the sub-word process reversible.

Transformer-Chinese word segmenter, using CPM-2 to Chinese vocabulary, it contains 26240 symbols and is trained by the Tsinghua team using 2.3TB of cleaned Chinese data. The CPM-2 vocabulary is modified based on the Chinese BPE. The original BPE inserts many redundant space marks "_" in the word segmentation sequence. The Tsinghua team replaces the sentence segmentation device with a combination of the word segmentation device and the stammering word segmentation, and deletes it Inserted spaces. Since it does not matter whether the symbols in the vocabulary appear at the beginning of the word, tags like "happy" (happy) and "_happy" (_happy) are merged into a single tag "happy" to simplify the vocabulary. Chinese data comes from multiple fields, including encyclopedias, novels, question and answer, scientific literature, e-books, news and reviews.

Transformer-English word segmentation, using CPM-2 to the English vocabulary, it contains 29,752 symbols, which is trained by the Tsinghua team using 300GB cleaned English data. The CPM-2 vocabulary is modified based on the English BPE. The original BPE inserts many redundant space marks "_" in the word segmentation sequence. The Tsinghua team replaced the sentence segmenter with the word segmenter combined with nltk word segmentation, and deleted Inserted spaces. English data comes from multiple fields, including encyclopedias, novels, Q&A, scientific literature, e-books, news, and reviews.

BERT-English word segmentation uses BERT's English vocabulary, which contains 30,000 symbols. It was trained by the Google team using WordPiece embedding on BooksCorpus (800M words) and English Wikipedia (2500M words).

The BERT-Chinese word segmentation period uses

BERT's Chinese vocabulary, which contains 21128 symbols and was trained by the Harbin Institute of Technology team using 13.6 million lines of Chinese Wikidata. In the process of generating the BERT vocabulary, Harbin Institute of Technology downloaded the latest Wikipediadump and used WikiExtractor.py to preprocess it according to the recommendations of Devlin *et al*., resulting in 1,307 extracted files, and used LTP for Chinese word segmentation.

English corpus, using the 800GB corpus collected by The Pile. In the process of using The Pile, due to the limitation of computing resources, not all the data was selected. Only the four blocks of 02, 03, 04, 17 about 213G data were used and removed. Two types of corpus, StackExchange and Github, which contain a lot of impurities, are included, and the entire dataset is about 200G. The Pile dataset is a new dataset collected by EleutherAI from PubMedCentral, ArXiv, GitHub, FreeLaw Project, StackExchange, U.S. Patent and Trademark Office, PubMed, UbuntuIRC, HackerNews, YouTube, PhilPapers and NIHExPorter, and introduces OpenWebText2 and BookCorpus2, they They are extensions of the original OpenWebText and BookCorpus datasets.

Chinese corpus, using 3TB Chinese corpus collected by Wudao. In the process of using Wudao corpus, due to the limitation of computing resources, 200G corpus was used. The Wudao dataset uses 3 billion web pages as the original data source to extract text content from web pages with high text density. Evaluate the quality of each data source before extracting the text, and ignore web pages with text density below 70%. The author uses the simhash algorithm to remove those repetitive content, and filters out those that do not contain meaningful sentences, few words, sensitive information, high-frequency garbled characters, fragments containing more than ten consecutive non-Chinese characters, HTML, Cascading Style Sheets (CSS) and Javascript Web page, remove private information, use punctuation marks (ie period, exclamation mark, question mark, ellipsis) to split the extracted text and delete the last paragraph, which may sometimes be incomplete. Convert traditional characters to simplified characters, and remove abnormal symbols (ie Emoticons, signs, etc.), delete all spaces in each sentence.

Chinese dialogue data, from the STC[10]-680M corpus dataset. A corpus of approximately 4.4 million conversations on Weibo. In order to build this million-scale dataset, the STC dataset first grabs hundreds of millions of response pairs, and then filters out potential responses by deleting trivial responses such as "wow". Advertisements, and delete the content after the first 30 responses to keep the theme consistent to clean up the original data.

name	corpus	language
BERT-C	200G	CN
BERT-E-S	200G	EN
BERT-E-M	200G	EN
BERT-E-L	200G	EN
BERT-E-X	200G	EN
BERT-E-E	200G	EN
BERT-X-CN-S	200G	CN
BERT-X-EN-S	200G	EN
BERT-X-EN-M	200G	EN
CPM-X-S	200G	CN
CPM-X-M	200G	CN
CPM-X-L	200G	CN
EVA-X	684M	CN
EPM-X-S	200G	EN
EPM-X-M	200G	EN
EPM-X-L	200G	EN
EPM-X-X	200G	EN
CPM-2-X-S	200G	CN
CPM-2-X-M	200G	CN
EPM-2-X-S	200G	EN

Table 1. Corpus and language for the FPM models.

6.2. Training Optimization and Hyperparameters

In order to effectively train the model, in some experiments, the mixed precision training and dynamic loss scaling are removed to use the tensor core of A100, but in some experiments, the training fails to converge due to accuracy reasons, and the mixed precision training is removed. First initialize the weights with a simple normal distribution, and then scale the weights immediately before the residual layer, where N is the number of transformer layers composed of self-attention and MLP blocks. For the optimizer, use Adam with a weight decay of 0.01. In addition, a 1.0 global gradient norm crop is used to improve the stability of training large models. In all cases, use a dropout of 0.1. Finally, in order to better manage the memory footprint, activation checkpoints are used after each transformer layer.

For the GPT-2 model, all training is performed using 1024 symbol sequences, the batch size is 512, and 300k iterations are performed. The 1.5e-4 learning rate is preheated for 3k iterations, and then a single-loop cosine decay is performed in the remaining iterations. Stop attenuation at the minimum learning rate of 1e-5.

For the BERT model, the training process described in the original text is mainly followed. Using the original BERT dictionary, the vocabulary size is 30,522. In addition, follow the suggested sentence order prediction to reposition the next sentence prediction head, and use the whole word n-gram mask. For all cases, set the batch size to 1024 and use a learning rate of 1.0e-4, warm up in 10,000 iterations

model-name	param	n-layer
BERT-C	330M	24
BERT-E-S	687.5M	50
BERT-E-M	825M	60
BERT-E-L	962.5M	70
BERT-E-X	1.1B	80
BERT-E-E	1.24B	90
BERT-X-CN-S	495M	36
BERT-X-EN-S	495M	36
BERT-X-EN-M	687.5M	48
CPM-X-S	2.9B	36
CPM-X-M	5.1B	64
CPM-X-L	10.3B	128
EVA-X	2.9B	36
EPM-X-S	2.9B	36
EPM-X-M	4B	50
EPM-X-L	5.1B	64
EPM-X-X	6.4B	80
CPM-2-X-S	2.9B	12
CPM-2-X-M	5.6B	24
EPM-2-X-S	2.9B	12

Table 2. The FPM models parameters and layers.

and decay linearly in the remaining iterations. Other training parameters remain unchanged.

7. Experiments

All experiments used up to 2 DGX servers (a total of 16 A100 SXM3 40GB GPUs). The infrastructure is optimized for multi-node deep learning applications. Through NVSwitch, a bandwidth of 300GB/sec is achieved between GPUs in the server, and a bandwidth of 10GB/sec is achieved between servers that use 1 InfiniBand adapter per server. . For Chinese, using the GPT-2 model, a 10.3 billion parameter language model was trained on the Chinese dataset, and, in particular, a 2.9 billion parameter language model based on dialogue data was trained; the BERT model was trained on the Chinese dataset with 495 million parameters; the Transformer model has trained a language model with 5.6 billion parameters on the Chinese dataset. In English, corresponding training work has also been done. Using the GPT-2 model, a language model with 6.4 billion parameters was trained on the English dataset; the BERT model trained a language model with 1.24 billion parameters on the English dataset, and in particular, it trained a 688 million parameter based on single card training technology Language model; Transformer model trained a language model with 2.9 billion parameters on the English dataset.

7.1. EPM-X

Based on the Transformer model, we built the encoding and decoding language model EPM-2-X, using the Transformer-English tokenizer, and trained a language model with 2.9 billion parameters. It has a 12-layer network structure, 6 encoding layers and 6 decoding layers.

7.2. EPM-2-X

Based on the Transformer model, we built the encoding and decoding language model EPM-2-X, using the Transformer-English tokenizer, and trained a language model with 2.9 billion parameters. It has a 12-layer network structure, 6 encoding layers and 6 decoding layers.

7.3. BERT-E

Based on the BERT model, we built the coded language model BERT-E. Using the BERT-English word segmenter, we trained 5 models with different layers. The largest model is a language model with 1.24 billion parameters. It has 90 layers of Transformer coding layers. The network is cascaded, and we named it BERT-EE.

7.4. BERT-X-EN

Based on the BERT model, we built the coded language model BERT-X-EN. Using the BERT-English word segmentation, we trained 2 models with different layers. The largest model is 690 million parameters, including 48 layers of Transformer coding layer network stack We named it BERT-X-EN-M.

7.5. CPM-X

Based on the GPT-2 model, we built a generative language model CPM-X, using GPT-Chinese word segmentation, trained 3 models with different levels, the largest model is a 10.3 billion parameter language model, it has 128 layers Transformer decoding layer network is stacked, we named it CPM-XL. Based on the GPT-2 model, we built a generative language model EVA-X, using GPT-Chinese word segmentation, using STC dialogue data, a language model of 2.9 billion parameters, it has 36 layers of Transformer decoding layer network stacked, we will It was named EVA-X.

7.6. CPM-2-X

Based on the Transformer model, we built the codec language model CPM-2-X, we trained 2 models, the largest of which uses Transformer-Chinese word segmentation, trained a 5.6 billion parameter language model, it has a 24-layer network structure , 12 encoding layers and 12 decoding layers, we named it CPM-2-XM.

model-name	d-n-hidden	n-heads	d-h-hidden
BERT-C	1024	16	64
BERT-E-S	1024	16	64
BERT-E-M	1024	16	64
BERT-E-L	1024	16	64
BERT-E-X	1024	16	64
BERT-E-E	1024	16	64
BERT-X-CN-S	1024	16	64
BERT-X-EN-S	1024	16	64
BERT-X-EN-M	1024	16	64
CPM-X-S	2560	32	80
CPM-X-M	2560	32	80
CPM-X-L	2560	32	80
EVA-X	2560	32	80
EPM-X-S	2560	32	80
EPM-X-M	2560	32	80
EPM-X-L	2560	32	80
EPM-X-X	2560	32	80
CPM-2-X-S	4096	64	64
CPM-2-X-M	4096	64	64
EPM-2-X-S	4096	64	64

Table 3. The FPM models' archetecture configurations.

model-name	time	step	gpu	Flops
BERT-C	81h	750k	8	727EFlops
BERT-E-S	34h	500k	8	305EFlops
BERT-E-M	38h51m	500k	8	349EFlops
BERT-E-L	45h38m	500k	8	410EFlops
BERT-E-X	54h24m	500k	8	488EFlops
BERT-E-E	63h	500k	8	566EFlops
BERT-X-CN-S	96h	880k	1	107EFlops
BERT-X-EN-S	315h	2800k	1	353EFlops
BERT-X-EN-M	315h	2800k	1	353EFlops
CPM-X-S	3h	10k	8	26.9EFlops
CPM-X-M	12h	60k	8	108EFlops
CPM-X-L	24h	100k	8	216EFlops
EVA-X	25h	160k	2	54EFlops
EPM-X-S	20h	320k	4	92EFlops
EPM-X-M	24h	320k	4	110EFlops
EPM-X-L	27h	320k	4	121EFlops
EPM-X-X	30h	320k	4	135EFlops
CPM-2-X-S	60h	200k	2	135EFlops
CPM-2-X-M	138h	80k	8	1240EFlops
EPM-2-X-S	110h	200k	2	247EFlops

Table 4. The cost for the FPM models.

7.7. BERT-C

Based on the BERT model, we built the coded language model BERT-C. Using the BERT-Chinese word segmenter, we trained a language model with 330 million parameters. It has a 24-layer Transformer coding layer network cascaded, and we named it BERT-C.

7.8. BERT-X-CN

Based on the BERT model, we built the coded language model BERT-X-CN. Using the BERT-Chinese word segmenter, we trained a language model with 495 million parameters, including 36 layers of Transformer coding layers. We named it BERT-X-CN-S.

8. Evaluation

In the evaluation stage, we evaluated the QQP classification task evaluated by GLUE on the BERT-E-S model.

8.1. GLUE-QQP

We use a Nvidia 3090 graphics card to fine-tune the QQP dataset on the BERT-E-S model. The BERT-E-S model has 50 layers, 1024 hidden dimensions, 16 attention heads, each head contains 64 hidden layers, and 687 million parameters. In the QQP classification task evaluated by GLUE, after 18 hours of fine-tuning 270,000 steps on the 109M corpus, the accuracy rate of 78.95% exceeded the accuracy of BERT-Large of 72.1%, an increase of 6.85%. Compared with the current 75.2% accuracy rate of ERNIE, the first place in the GLUE evaluation, this is an increase of 3.75%.

8.2. CLUE1.0-TNEWS

We use an Nvidia 3090 graphics card to fine-tune the TNEWS dataset using the BERT-C model. The BERT-C model has 24 layers, 1024 hidden dimensions, 16 attention heads, each head contains 64 hidden layers, and 330 million parameters. In the TNEWS classification task evaluated by CLUE, after fine-tuning the 9.7M corpus for 8 hours and 70,000 steps, the accuracy rate of 59.99% surpassed the accuracy rate of 59.46% of ALBERT-xxlarge, an increase of 0.53%.

9. Discussion

We will discuss from the perspective of the language, scale, network configuration, and cost of the training models.

9.1. Language

From Table-6.1, Chinese is the single country with the largest number of users, and English is the most widely used language. This job has trained a large number of Chinese and English corpora and conducted detailed training.

9.2. Scale

From Table-6.2, the models with the least number of layers are CPM-2-X and EPM-2-X, with only 12 layers. The

model with the largest number of layers is CPM-X, with 128 layers trained. From the perspective of the difficulty of training, the GPT structure is easier to cascade the number of layers; while the Transformer codec structure is not easy to increase the number of layers, and it is easier to touch the memory limit of the hardware.

From the perspective of model parameters, the model with the least amount of parameters is BERT-C, with only 330 million, and the largest parameter is CPM-X-L, with parameters reaching 10.3 billion.

9.3. Architecture Configuration

From Table-7.8, for BERT, GPT, and Transformer, the network structure parameters used in this work are fixed, and the parameters in the original paper are used more without major modifications.

9.4. Cost

It can be seen from Table-8.2 that in terms of time consumption, the shortest training time is CPM-XS, and the longest model is CPM-X-EN; from the training step number is CPM-XS, only 100,000 steps, the most training It is BERT-X-EN with 2.8 million steps. From the perspective of computing power, the least computing power is CPM-XS, which uses 26.9EFlops, and the most computing power is CPM-2-XM, which uses 1240EFlops.

10. Conclusion

In this work, the latest technology for training super-large converter models is used, and a simple and efficient intra-layer model parallel method is used to train dozens of converter models with hundreds of millions or even tens of billions of parameters.

For Chinese, using the GPT-2 model, a 10.3 billion parameter language model was trained on the Chinese dataset, and, in particular, a 2.9 billion parameter language model based on dialogue data was trained; the BERT model was trained on the Chinese dataset with 495 million parameters; the Transformer model has trained a language model with 5.6 billion parameters on the Chinese dataset.

In English, corresponding training work has also been done. Using the GPT-2 model, a language model with 6.4 billion parameters was trained on the English dataset; the BERT model trained a language model with 1.24 billion parameters on the English dataset, and in particular, it trained a 688 million parameter based on single card training technology Language model; Transformer model trained a language model with 2.9 billion parameters on the English dataset.

In the TNEWS classification task evaluated by CLUE, the BERT-C model exceeded the 59.46% accuracy of ALBERT-xxlarge with an accuracy rate of 59.99%, an increase of 0.53%.

In the QQP classification task evaluated by GLUE, the accuracy rate of 78.95% surpassed the accuracy rate of BERT-Large of 72.1%, an increase of 6.85%. Compared with the current accuracy rate of ERNIE, the first place in the GLUE evaluation of 75.2%, an increase of 3.75%.

References

[1] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 2

[2] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*, 2019. 2

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. 1

[4] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020. 3

[5] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019. 2

[6] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019. 1

[7] Junyang Lin, Rui Men, An Yang, Chang Zhou, Ming Ding, Yichang Zhang, Peng Wang, Ang Wang, Le Jiang, Xianyan Jia, et al. M6: A chinese multimodal pretrainer. *arXiv preprint arXiv:2103.00823*, 2021. 2

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 2

[9] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *Technical report*, 2020. 1

[10] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*, 2015. 4

[11] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, 2018. 1

[12] Ben Wang. Mesh-Transformer-JAX: Model-Parallel Implementation of Transformer Language Model with JAX. https://github.com/kingoflolz/mesh-transformer-jax, May 2021. 2

[13] Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, et al. Clue: A chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, 2020. 1

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

[15] Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, et al. Pangu-$alpha$: Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *arXiv preprint arXiv:2104.12369*, 2021. 2

[16] Zhengyan Zhang, Yuxian Gu, Xu Han, Shengqi Chen, Chaojun Xiao, Zhenbo Sun, Yuan Yao, Fanchao Qi, Jian Guan, Pei Ke, et al. Cpm-2: Large-scale cost-effective pre-trained language models. *arXiv preprint arXiv:2106.10715*, 2021. 2

[17] Zhengyan Zhang, Xu Han, Hao Zhou, Pei Ke, Yuxian Gu, Deming Ye, Yujia Qin, Yusheng Su, Haozhe Ji, Jian Guan, et al. Cpm: A large-scale generative chinese pre-trained language model. *AI Open*, 2:93–99, 2021. 2