Fengshenbang-LM

发行版本 *1.0*

IDEA/CCNL

2022 年 12 月 17 日

Contents

CHAPTER 1

二郎神系列

1.1 Erlangshen-DeBERTa-v2-97M-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.1.1 简介 Brief Introduction

善于处理 NLU 任务，采用全词掩码的，中文版的 0.97 亿参数 DeBERTa-v2-Base。

Good at solving NLU tasks, adopting Whole Word Masking, Chinese DeBERTa-v2-Base with 97M parameters.

1.1.2 模型分类 Model Taxonomy

1.1.3 模型信息 Model Information

参考论文：DeBERTa: Decoding-enhanced BERT with Disentangled Attention

为了得到一个中文版的 DeBERTa-v2-Base（97M），我们用悟道语料库 (180G 版本) 进行预训练。我们在 MLM 中使用了全词掩码 (wwm) 的方式。具体地，我们在预训练阶段中使用了封神框架大概花费了 24 张 A100 约 7 天。

To get a Chinese DeBERTa-v2-Base (97M), we use WuDao Corpora (180 GB version) for pre-training. We employ the Whole Word Masking (wwm) in MLM. Specifically, we use the fengshen framework in the pre-training phase which cost

about 7 days with 24 A100 GPUs.

下游任务 Performance

我们展示了下列下游任务的结果（dev 集）：

We present the results (dev set) on the following tasks:

1.1.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-DeBERTa-v2-97M-Chinese

加载模型 Loading Models

```
from transformers import AutoModelForMaskedLM, AutoTokenizer, FillMaskPipeline
import torch

tokenizer=AutoTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-DeBERTa-v2-97M-Chinese',
↪ use_fast=False)
model=AutoModelForMaskedLM.from_pretrained('IDEA-CCNL/Erlangshen-DeBERTa-v2-97M-
↪Chinese')
text = '生活的真谛是[MASK]。'
fillmask_pipe = FillMaskPipeline(model, tokenizer, device=7)
print(fillmask_pipe(text, top_k=10))
```

1.1.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
```

(续下页)

(接上页)

```
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.2 Erlangshen-DeBERTa-v2-97M-CWS-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.2.1 简介 Brief Introduction

善于处理 NLU 任务，采用中文分词的，中文版的 0.97 亿参数 DeBERTa-v2-Base。

Good at solving NLU tasks, adopting Chinese Word Segmentation (CWS), Chinese DeBERTa-v2-Base with 97M parameters.

1.2.2 模型分类 Model Taxonomy

1.2.3 模型信息 Model Information

参考论文：DeBERTa: Decoding-enhanced BERT with Disentangled Attention

为了得到一个中文版的 DeBERTa-v2-Base（97M），我们用悟道语料库 (180G 版本) 进行预训练。我们使用了中文分词。具体地，我们在预训练阶段中使用了封神框架大概花费了 24 张 A100 约 7 天。

To get a Chinese DeBERTa-v2-Base (97M), we use WuDao Corpora (180 GB version) for pre-training. We employ Chinese Word Segmentation (CWS). Specifically, we use the fengshen framework in the pre-training phase which cost about 7 days with 24 A100 GPUs.

1.2.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-DeBERTa-v2-97M-CWS-Chinese

加载模型 Loading Models

```
from transformers import AutoModelForMaskedLM, AutoTokenizer, FillMaskPipeline
import torch

tokenizer=AutoTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-DeBERTa-v2-97M-CWS-
↪Chinese', use_fast=False)
model=AutoModelForMaskedLM.from_pretrained('IDEA-CCNL/Erlangshen-DeBERTa-v2-97M-CWS-
↪Chinese')
text = '生活的真谛是[MASK]。'
fillmask_pipe = FillMaskPipeline(model, tokenizer, device=7)
print(fillmask_pipe(text, top_k=10))
```

1.2.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
```

(续下页)

(接上页)

```
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.3 Erlangshen-DeBERTa-v2-186M-Chinese-SentencePiece

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.3.1 简介 Brief Introduction

善于处理 NLU 任务，采用 sentence piece 分词的，中文版的 1.86 亿参数 DeBERTa-v2。

Good at solving NLU tasks, adopting sentence piece, Chinese DeBERTa-v2 with 186M parameters.

1.3.2 模型分类 Model Taxonomy

1.3.3 模型信息 Model Information

为了得到一个中文版的 DeBERTa-v2（186M），我们用悟道语料库 (180G 版本) 进行预训练。我们使用了 Sentence Piece 的方式分词（词表大小：约 128000）。具体地，我们在预训练阶段中使用了封神框架大概花费了 8 张 3090TI（24G）约 21 天。

To get a Chinese DeBERTa-v2 (186M), we use WuDao Corpora (180 GB version) for pre-training. We employ the sentence piece as the tokenizer (vocabulary size: around 128,000). Specifically, we use the fengshen framework in the pre-training phase which cost about 21 days with 8 3090TI（24G）GPUs.

下游效果 Performance

我们展示了下列下游任务的结果（dev 集）：

We present the results (dev set) on the following tasks:

1.3.4 使用 Usage

模型下载地址

Huggingface 地址：Erlangshen-DeBERTa-v2-186M-Chinese-SentencePiece

加载模型 Loading Models

```
from transformers import AutoModelForMaskedLM, AutoTokenizer, FillMaskPipeline
import torch

tokenizer=AutoTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-DeBERTa-v2-186M-Chinese-
↪SentencePiece', use_fast=False)
model=AutoModelForMaskedLM.from_pretrained('IDEA-CCNL/Erlangshen-DeBERTa-v2-186M-
↪Chinese-SentencePiece')
text = '中国首都位于<mask>。'
fillmask_pipe = FillMaskPipeline(model, tokenizer)
print(fillmask_pipe(text, top_k=10))
```

1.3.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
```

(续下页)

(接上页)

```
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.4 Erlangshen-DeBERTa-v2-320M-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.4.1 简介 Brief Introduction

善于处理 NLU 任务，采用全词掩码的，中文版的 3.2 亿参数 DeBERTa-v2-Large。

Good at solving NLU tasks, adopting Whole Word Masking, Chinese DeBERTa-v2-Large with 320M parameters.

1.4.2 模型分类 Model Taxonomy

1.4.3 模型信息 Model Information

参考论文：DeBERTa: Decoding-enhanced BERT with Disentangled Attention

为了得到一个中文版的 DeBERTa-v2-large（320M），我们用悟道语料库 (180G 版本) 进行预训练。我们在 MLM 中使用了全词掩码 (wwm) 的方式。具体地，我们在预训练阶段中使用了封神框架大概花费了 8 张 A100（80G）约 7 天。

To get a Chinese DeBERTa-v2-large (320M), we use WuDao Corpora (180 GB version) for pre-training. We employ the Whole Word Masking (wwm) in MLM. Specifically, we use the fengshen framework in the pre-training phase which cost about 7 days with 8 A100（80G）GPUs.

下游任务 Performance

我们展示了下列下游任务的结果（dev 集）：

We present the results (dev set) on the following tasks:

1.4.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-DeBERTa-v2-320M-Chinese

加载模型 Loading Models

```
from transformers import AutoModelForMaskedLM, AutoTokenizer, FillMaskPipeline
import torch

tokenizer=AutoTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-DeBERTa-v2-320M-Chinese
↪', use_fast=False)
model=AutoModelForMaskedLM.from_pretrained('IDEA-CCNL/Erlangshen-DeBERTa-v2-320M-
↪Chinese')
text = '桂林是世界闻名的旅游城市,它有[MASK]江。'
fillmask_pipe = FillMaskPipeline(model, tokenizer, device=0)
print(fillmask_pipe(text, top_k=10))
```

1.4.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
```

(续下页)

(接上页)

```
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.5 Erlangshen-DeBERTa-v2-710M-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.5.1 简介 Brief Introduction

善于处理 NLU 任务，采用全词掩码的，中文版的 7.1 亿参数 DeBERTa-v2-XLarge。

Good at solving NLU tasks, adopting Whole Word Masking, Chinese DeBERTa-v2-XLarge with 710M parameters.

1.5.2 模型分类 Model Taxonomy

1.5.3 模型信息 Model Information

参考论文：DeBERTa: Decoding-enhanced BERT with Disentangled Attention

为了得到一个中文版的 DeBERTa-v2-xlarge（710M），我们用悟道语料库 (180G 版本) 进行预训练。我们在 MLM 中使用了全词掩码 (wwm) 的方式。具体地，我们在预训练阶段中使用了封神框架大概花费了 24 张 A100（40G）约 21 天。

To get a Chinese DeBERTa-v2-xlarge (710M), we use WuDao Corpora (180 GB version) for pre-training. We employ the Whole Word Masking (wwm) in MLM. Specifically, we use the fengshen framework in the pre-training phase which cost about 21 days with 24 A100（40G）GPUs.

下游任务 Performance

我们展示了下列下游任务的结果：

We present the results on the following tasks:

1.5.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-DeBERTa-v2-710M-Chinese

加载模型 Loading Models

```
from transformers import AutoModelForMaskedLM, AutoTokenizer, FillMaskPipeline
import torch

tokenizer=AutoTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-DeBERTa-v2-710M-Chinese
↪', use_fast=False)
model=AutoModelForMaskedLM.from_pretrained('IDEA-CCNL/Erlangshen-DeBERTa-v2-710M-
↪Chinese')
text = '生活的真谛是[MASK]。'
fillmask_pipe = FillMaskPipeline(model, tokenizer, device=-1)
print(fillmask_pipe(text, top_k=10))
```

1.5.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
```

(续下页)

(接上页)

```
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.6 Erlangshen-Longformer-110M

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.6.1 简介 Brief Introduction

善于处理长文本，采用旋转位置编码的中文版 1.1 亿参数的 Longformer-Base。

The Chinese Longformer-Base (110M), which uses rotating positional encoding, is adept at handling lengthy text.

1.6.2 模型分类 Model Taxonomy

1.6.3 模型信息 Model Information

遵循 Longformer-Base 的设计，我们基于chinese_roformer_L-12_H-768_A-12，在悟道语料库 (180 GB 版本) 上进行了继续预训练。特别的，我们采用旋转位置嵌入 (RoPE) 来避免预训练语料库的不均匀序列长度问题。

Following the design of Longformer-Base, we performed continual pre-training on the WuDao corpus (180 GB) based on chinese_roformer_L-12_H-768_A-12. Particularly, we employed rotational position embedding (RoPE) to avoid the uneven sequence length of the pre-trained corpus.

1.6.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-Longformer-110M

加载模型 Loading Models

因为transformers库中是没有 Longformer-Base 相关的模型结构的，所以你可以在我们的Fengshenbang-LM中找到并且运行代码。

Since there is no structure of Longformer-Base in transformers library, you can find the structure of Longformer-Base and run the codes in Fengshenbang-LM.

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
```

加载模型 Loading Models

```
from fengshen import LongformerModel
from fengshen import LongformerConfig
from transformers import BertTokenizer

tokenizer = BertTokenizer.from_pretrained("IDEA-CCNL/Erlangshen-Longformer-110M")
config = LongformerConfig.from_pretrained("IDEA-CCNL/Erlangshen-Longformer-110M")
model = LongformerModel.from_pretrained("IDEA-CCNL/Erlangshen-Longformer-110M")
```

1.6.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.7 Erlangshen-Longformer-330M

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.7.1 简介 Brief Introduction

善于处理长文本，采用旋转位置编码的中文版 3.3 亿参数的 Longformer-large

The Chinese Longformer-large (330M), which uses rotating positional encoding, is adept at handling lengthy text.

1.7.2 模型分类 Model Taxonomy

1.7.3 模型信息 Model Information

遵循 Longformer-large 的设计，我们基于chinese_roformer_L-12_H-768_A-12，在悟道语料库 (180 GB 版本) 上进行了继续预训练。特别的，我们采用旋转位置嵌入 (RoPE) 来避免预训练语料库的不均匀序列长度问题。

Following the design of Longformer-large, we performed continual pre-training on the WuDao corpus (180 GB) based on chinese_roformer_L-12_H-768_A-12. Particularly, we employed rotational position embedding (RoPE) to avoid the uneven sequence length of the pre-trained corpus.

1.7.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-Longformer-330M

加载模型 Loading Models

因为transformers库中是没有 Longformer-large 相关的模型结构的，所以你可以在我们的Fengshenbang-LM中找到并且运行代码。

Since there is no structure of Longformer-large in transformers library, you can find the structure of Longformer-base and run the codes in Fengshenbang-LM.

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
```

```
from fengshen import LongformerModel
from fengshen import LongformerConfig
from transformers import BertTokenizer

tokenizer = BertTokenizer.from_pretrained("IDEA-CCNL/Erlangshen-Longformer-330M")
config = LongformerConfig.from_pretrained("IDEA-CCNL/Erlangshen-Longformer-330M")
model = LongformerModel.from_pretrained("IDEA-CCNL/Erlangshen-Longformer-330M")
```

1.7.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
```

(续下页)

(接上页)

```
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.8 Erlangshen-MegatronBert-1.3B-NLI

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.8.1 简介 Brief Introduction

2021 年登顶 FewCLUE 和 ZeroCLUE 的中文 BERT，在数个推理任务微调后的版本

This is the fine-tuned version of the Chinese BERT model on several NLI datasets, which topped FewCLUE and ZeroCLUE benchmark in 2021

1.8.2 模型分类 Model Taxonomy

1.8.3 模型信息 Model Information

基于Erlangshen-MegatronBert-1.3B，我们在收集的 4 个中文领域的 NLI（自然语言推理）数据集，总计 1014787 个样本上微调了一个 NLI 版本。

Based on Erlangshen-MegatronBert-1.3B, we fine-tuned a NLI version on 4 Chinese Natural Language Inference (NLI) datasets, with totaling 1,014,787 samples.

下游效果 Performance

1.8.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-MegatronBert-1.3B-NLI

加载模型 Loading Models

```
from transformers import AutoModelForSequenceClassification
from transformers import BertTokenizer
import torch
tokenizer=BertTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-MegatronBert-1.3B-NLI')
model=AutoModelForSequenceClassification.from_pretrained('IDEA-CCNL/Erlangshen-
↪MegatronBert-1.3B-NLI')
texta='今天的饭不好吃'
textb='今天心情不好'
output=model(torch.tensor([tokenizer.encode(texta,textb)]))
print(torch.nn.functional.softmax(output.logits,dim=-1))
```

数据样本示例 Data Examples

```
{
  "texta": "身上裹一件工厂发的棉大衣,手插在袖筒里",
  "textb": "身上至少一件衣服",
  "label": 2,
  "id": 0
 }
```

标签映射：模型输出 0 表示两个句子矛盾，1 表示没有关系，2 表示蕴含关系

```
"id2label": {
    "0": "CONTRADICTION",
    "1": "NEUTRAL",
    "2": "ENTAILMENT"
  },
```

1.8.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
```

(续下页)

(接上页)

```
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.9 Erlangshen-MegatronBert-1.3B-Semtiment

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.9.1 简介 Brief Introduction

2021 年登顶 FewCLUE 和 ZeroCLUE 的中文 BERT，在数个情感分析任务微调后的版本。

This is the fine-tuned version of the Chinese BERT model on several sentiment analysis datasets, which topped FewCLUE and ZeroCLUE benchmark in 2021.

1.9.2 模型分类 Model Taxonomy

1.9.3 模型信息 Model Information

基于Erlangshen-MegatronBert-1.3B，我们在收集的 8 个中文领域的情感分析数据集，总计 227347 个样本上微调了一个 Semtiment 版本。

Based on Erlangshen-MegatronBert-1.3B, we fine-tuned a sentiment analysis version on 8 Chinese sentiment analysis datasets, with totaling 227,347 samples.

下游效果 Performance

1.9.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-MegatronBert-1.3B-Sentiment

加载模型 Loading Models

```
from transformers import AutoModelForSequenceClassification
from transformers import BertTokenizer
import torch

tokenizer=BertTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-MegatronBert-1.3B-
↪Sentiment')
model=AutoModelForSequenceClassification.from_pretrained('IDEA-CCNL/Erlangshen-
↪MegatronBert-1.3B-Sentiment')

text='今天心情不好'

output=model(torch.tensor([tokenizer.encode(text)]))
print(torch.nn.functional.softmax(output.logits,dim=-1))
```

数据样本示例 Data Examples

```
{
  "texta": "外形还OK,用了2天了在VISTA下玩游戏还行的.发热量有时大有时小不知道为什么,
↪不过总体上来说还不是很大,4600买的还送个大礼包.",
  "textb": "",
  "label": 1,
  "id": "33"
    }
```

标签映射：模型输出 0 表示消极，输出 1 表示积极

```
"id2label":{
      "0":"Negative",
      "1":"Positive"
       }
```

1.9.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.10 Erlangshen-MegatronBert-1.3B-Similarity

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.10.1 简介 Brief Introduction

2021 年登顶 FewCLUE 和 ZeroCLUE 的中文 BERT，在数个相似度任务上微调后的版本。

This is the fine-tuned version of the Chinese BERT model on several similarity datasets, which topped FewCLUE and ZeroCLUE benchmark in 2021.

1.10.2 模型分类 Model Taxonomy

1.10.3 模型信息 Model Information

基于Erlangshen-MegatronBert-1.3B，我们在收集的 20 个中文领域的改写数据集，总计 2,773,880 个样本上微调了一个 Similarity 版本。

Based on Erlangshen-MegatronBert-1.3B, we fine-tuned a similarity version on 20 Chinese paraphrase datasets, with totaling 2,773,880 samples.

成就 Achievements

我们于 2022 年 7 月 10 日登顶 CLUE 语义匹配榜，详情见Towards No.1 in CLUE Semantic Matching Challenge: Pre-trained Language Model Erlangshen with Propensity-Corrected Loss。

We topped the CLUE benchmark semantic matching task on July 10, 2022, as detailed in Towards No.1 in CLUE Semantic Matching Challenge: Pre-trained Language Model Erlangshen with Propensity-Corrected Loss.

下游效果 Performance

1.10.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-MegatronBert-1.3B-Similarity

加载模型 Loading Models

```
from transformers import AutoModelForSequenceClassification
from transformers import BertTokenizer
import torch

tokenizer=BertTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-MegatronBert-1.3B-
↪Similarity')
model=AutoModelForSequenceClassification.from_pretrained('IDEA-CCNL/Erlangshen-
↪MegatronBert-1.3B-Similarity')

texta='今天的饭不好吃'
textb='今天心情不好'

output=model(torch.tensor([tokenizer.encode(texta,textb)]))
print(torch.nn.functional.softmax(output.logits,dim=-1))
```

数据样本示例

```
{
  "texta": "可以🅵其他银行卡吗？",
  "textb": "分期的如何用别的银行卡还钱",
  "label": 1,
  "id": 0
  }
```

标签映射：模型输出 0 表示不相似，输出 1 表示相似

```
"id2label":{
    "0":"not similarity",
    "1":"similarity"
     }
```

1.10.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的对该模型的论文：

If you are using the resource for your work, please cite the our paper for this model:

```
@article{fengshenbang/erlangshen-megatronbert-sim,
  author    = {Junjie Wang and
               Yuxiang Zhang and
               Ping Yang and
               Ruyi Gan},
  title     = {Towards No.1 in {CLUE} Semantic Matching Challenge: Pre-trained␣
↪Language
               Model Erlangshen with Propensity-Corrected Loss},
  journal   = {CoRR},
  volume    = {abs/2208.02959},
  year      = {2022}
}
```

如果您在您的工作中使用了我们的模型，也可以引用我们的总论文：

If you are using the resource for your work, please cite the our overview paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
```

(续下页)

(接上页)

```
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.11 Erlangshen-MegatronBert-3.9B-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.11.1 简介 Brief Introduction

善于处理 NLU 任务，现在最大的，拥有 39 亿的中文 BERT 模型。

Good at solving NLU tasks, the largest Chinese BERT (39B) currently.

1.11.2 模型分类 Model Taxonomy

1.11.3 模型信息 Model Information

Erlangshen-MegatronBert-3.9B-Chinese 是一个比Erlangshen-MegatronBert-1.3B拥有更多参数的版本（39 亿）。我们遵循原来的预训练方式在悟道数据集（300G 版本）上进行预训练。具体地，我们在预训练阶段中使用了封神框架大概花费了 64 张 A100（40G）约 30 天。

Erlangshen-MegatronBert-3.9B-Chinese (3.9B) is a larger version of Erlangshen-MegatronBert-1.3B. By following the original instructions, we apply WuDao Corpora (300 GB version) as the pretraining dataset. Specifically, we use the fengshen framework in the pre-training phase which cost about 30 days with 64 A100 (40G) GPUs.

更多信息 More Information

IDEA 研究院中文预训练模型二郎神登顶 FewCLUE 榜单

2021 年 11 月 10 日，Erlangshen-MegatronBERT-1.3B 在 FewCLUE 上取得第一。其中，它在 CHIDF(成语填空) 和 TNEWS(新闻分类) 子任务中的表现优于人类表现。此外，它在 CHIDF(成语填空), CSLDCP(学科文献分类), OCNLI(自然语言推理) 任务中均名列前茅。

On November 10, 2021, Erlangshen-MegatronBert-1.3B topped the FewCLUE benchmark. Among them, our Erlangshen outperformed human performance in CHIDF (idiom fill-in-the-blank) and TNEWS (news classification) subtasks. In addition, our Erlangshen ranked the top in CHIDF (idiom fill-in-the-blank), CSLDCP (subject literature classification), and OCNLI (natural language inference) tasks.

下游效果 Performance

下游中文任务的得分（没有做任何数据增强）：

Scores on downstream Chinese tasks (without any data augmentation):

1.11.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-MegatronBert-3.9B-Chinese

加载模型 Loading Models

```
from transformers import AutoModelForMaskedLM, AutoTokenizer, FillMaskPipeline
import torch

tokenizer=AutoTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-MegatronBert-3.9B-
↪Chinese', use_fast=False)
model=AutoModelForMaskedLM.from_pretrained('IDEA-CCNL/Erlangshen-MegatronBert-3.9B-
↪Chinese')
text = '生活的真谛是[MASK]。'
fillmask_pipe = FillMaskPipeline(model, tokenizer)
print(fillmask_pipe(text, top_k=10))
```

1.11.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.12 Erlangshen-Roberta-110M-NLI

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.12.1 简介 Brief Introduction

中文的 RoBERTa-wwm-ext-base 在数个推理任务微调后的版本。

This is the fine-tuned version of the Chinese RoBERTa-wwm-ext-base model on several NLI datasets.

1.12.2 模型分类 Model Taxonomy

1.12.3 模型信息 Model Information

基于chinese-roberta-wwm-ext-base，我们在收集的 4 个中文领域的 NLI（自然语言推理）数据集，总计 1014787 个样本上微调了一个 NLI 版本。

Based on chinese-roberta-wwm-ext-base, we fine-tuned an NLI version on 4 Chinese Natural Language Inference (NLI) datasets, with totaling 1,014,787 samples.

下游效果 Performance

1.12.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-Roberta-110M-NLI

加载模型 Loading Models

```
from transformers import BertForSequenceClassification
from transformers import BertTokenizer
import torch

tokenizer=BertTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-110M-NLI')
model=BertForSequenceClassification.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-
↪110M-NLI')

texta='今天的饭不好吃'
textb='今天心情不好'

output=model(torch.tensor([tokenizer.encode(texta,textb)]))
print(torch.nn.functional.softmax(output.logits,dim=-1))
```

数据样本示例

```
{
  "texta": "身上裹一件工厂发的棉大衣,手插在袖筒里",
  "textb": "身上至少一件衣服",
  "label": 2,
  "id": 0
 }
```

标签映射：模型输出 0 表示两个句子矛盾，1 表示没有关系，2 表示蕴含关系

```
"id2label": {
    "0": "CONTRADICTION",
    "1": "NEUTRAL",
    "2": "ENTAILMENT"
  },
```

1.12.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.13 Erlangshen-Roberta-110M-Sentiment

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.13.1 简介 Brief Introduction

中文的 RoBERTa-wwm-ext-base 在数个情感分析任务微调后的版本

This is the fine-tuned version of the Chinese RoBERTa-wwm-ext-base model on several sentiment analysis datasets.

1.13.2 模型分类 Model Taxonomy

1.13.3 模型信息 Model Information

基于chinese-roberta-wwm-ext-base，我们在收集的 8 个中文领域的情感分析数据集，总计 227347 个样本上微调了一个 Semtiment 版本。

Based on chinese-roberta-wwm-ext-base, we fine-tuned a sentiment analysis version on 8 Chinese sentiment analysis datasets, with totaling 227,347 samples.

下游效果 Performance

1.13.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-Roberta-110M-Sentiment

加载模型 Loading Models

```
from transformers import BertForSequenceClassification
from transformers import BertTokenizer
import torch

tokenizer=BertTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-110M-Sentiment')
model=BertForSequenceClassification.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-
↪110M-Sentiment')

text='今天心情不好'
```

(续下页)

(接上页)

```
output=model(torch.tensor([tokenizer.encode(text)]))
print(torch.nn.functional.softmax(output.logits,dim=-1))
```

数据样本示例

```
{
  "texta": "外形还OK,用了2天了在VISTA下玩游戏还行的.发热量有时大有时小不知道为什么,
↪不过总体上来说还不是很大,4600买的还送个大礼包.",
  "textb": "",
  "label": 1,
  "id": "33"
    }
```

标签映射：模型输出 0 表示消极，输出 1 表示积极

```
"id2label":{
      "0":"Negative",
      "1":"Positive"
       }
```

1.13.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.14 Erlangshen-Roberta-110M-Similarity

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.14.1 简介 Brief Introduction

中文的 RoBERTa-wwm-ext-base 在数个相似度任务微调后的版本

This is the fine-tuned version of the Chinese RoBERTa-wwm-ext-base model on several similarity datasets.

1.14.2 模型分类 Model Taxonomy

1.14.3 模型信息 Model Information

基于chinese-roberta-wwm-ext-base，我们在收集的 20 个中文领域的改写数据集，总计 2773880 个样本上微调了一个 Similarity 版本。

Based on chinese-roberta-wwm-ext-base, we fine-tuned a similarity version on 20 Chinese paraphrase datasets, with totaling 2,773,880 samples.

下游效果 Performance

1.14.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-Roberta-110M-Similarity

加载模型 Loading Models

```
from transformers import BertForSequenceClassification
from transformers import BertTokenizer
import torch

tokenizer=BertTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-110M-Similarity
↪')
model=BertForSequenceClassification.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-
↪110M-Similarity')

texta='今天的饭不好吃'
textb='今天心情不好'

output=model(torch.tensor([tokenizer.encode(texta,textb)]))
print(torch.nn.functional.softmax(output.logits,dim=-1))
```

数据样本示例 Data Examples

```
{
  "texta": "可以🅵其他银行卡吗？",
  "textb": "分期的如何用别的银行卡还钱",
  "label": 1,
  "id": 0
  }
```

标签映射：模型输出 0 表示不相似，输出 1 表示相似

```
"id2label":{
    "0":"not similarity",
    "1":"similarity"
     }
```

1.14.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
```

(续下页)

(接上页)

```
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.15 Erlangshen-Roberta-330M-NLI

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.15.1 简介 Brief Introduction

中文的 RoBERTa-wwm-ext-large 在数个推理任务微调后的版本

This is the fine-tuned version of the Chinese RoBERTa-wwm-ext-large model on several NLI datasets.

1.15.2 模型分类 Model Taxonomy

1.15.3 模型信息 Model Information

基于chinese-roberta-wwm-ext-large，我们在收集的 4 个中文领域的 NLI（自然语言推理）数据集，总计 1014787 个样本上微调了一个 NLI 版本。

Based on chinese-roberta-wwm-ext-large, we fine-tuned an NLI version on 4 Chinese Natural Language Inference (NLI) datasets, with totaling 1,014,787 samples.

下游效果 Performance

1.15.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-Roberta-330M-NLI

加载模型 Loading Models

```
from transformers import BertForSequenceClassification
from transformers import BertTokenizer
import torch

tokenizer=BertTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-330M-NLI')
model=BertForSequenceClassification.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-
↪330M-NLI')

texta='今天的饭不好吃'
textb='今天心情不好'

output=model(torch.tensor([tokenizer.encode(texta,textb)]))
print(torch.nn.functional.softmax(output.logits,dim=-1))
```

数据样本示例 Data Examples

```
{
  "texta": "身上裹一件工厂发的棉大衣,手插在袖筒里",
  "textb": "身上至少一件衣服",
  "label": 2,
  "id": 0
 }
```

标签映射：模型输出 0 表示两个句子矛盾，1 表示没有关系，2 表示蕴含关系

```
"id2label": {
    "0": "CONTRADICTION",
    "1": "NEUTRAL",
    "2": "ENTAILMENT"
  },
```

1.15.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.16 Erlangshen-Roberta-330M-Sentiment

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.16.1 简介 Brief Introduction

中文的 RoBERTa-wwm-ext-large 在数个情感分析任务微调后的版本

This is the fine-tuned version of the Chinese RoBERTa-wwm-ext-large model on several sentiment analysis datasets.

1.16.2 模型分类 Model Taxonomy

1.16.3 模型信息 Model Information

基于chinese-roberta-wwm-ext-large，我们在收集的 8 个中文领域的情感分析数据集，总计 227347 个样本上微调了一个 Semtiment 版本。

Based on chinese-roberta-wwm-ext-large, we fine-tuned a sentiment analysis version on 8 Chinese sentiment analysis datasets, with totaling 227,347 samples.

下游效果 Performance

1.16.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-Roberta-330M-Sentiment

加载模型 Loading Models

```
from transformers import BertForSequenceClassification
from transformers import BertTokenizer
import torch

tokenizer=BertTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-330M-Sentiment')
model=BertForSequenceClassification.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-
↪330M-Sentiment')

text='今天心情不好'

output=model(torch.tensor([tokenizer.encode(text)]))
print(torch.nn.functional.softmax(output.logits,dim=-1))
```

数据样本示例 Data Examples

```
{
  "texta": "外形还OK,用了2天了在VISTA下玩游戏还行的.发热量有时大有时小不知道为什么,
↪不过总体上来说还不是很大,4600买的还送个大礼包.",
  "textb": "",
  "label": 1,
  "id": "33"
    }
```

标签映射：模型输出 0 表示消极，输出 1 表示积极

```
"id2label":{
      "0":"Negative",
      "1":"Positive"
       }
```

1.16.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.17 Erlangshen-Roberta-330M-Similarity

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.17.1 简介 Brief Introduction

中文的 RoBERTa-wwm-ext-large 在数个相似度任务微调后的版本

This is the fine-tuned version of the Chinese RoBERTa-wwm-ext-large model on several similarity datasets.

1.17.2 模型分类 Model Taxonomy

1.17.3 模型信息 Model Information

基于chinese-roberta-wwm-ext-large，我们在收集的 20 个中文领域的改写数据集，总计 2773880 个样本上微调了一个 Similarity 版本。

Based on chinese-roberta-wwm-ext-large, we fine-tuned a similarity version on 20 Chinese paraphrase datasets, with totaling 2,773,880 samples.

下游效果 Performance

1.17.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-Roberta-330M-Similarity

加载模型 Loading Models

```
from transformers import BertForSequenceClassification
from transformers import BertTokenizer
import torch

tokenizer=BertTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-330M-Similarity
↪')
model=BertForSequenceClassification.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-
↪330M-Similarity')

texta='今天的饭不好吃'
```

(续下页)

(接上页)

```
textb='今天心情不好'

output=model(torch.tensor([tokenizer.encode(texta,textb)]))
print(torch.nn.functional.softmax(output.logits,dim=-1))
```

数据样本示例 Data Examples

```
{
  "texta": "可以🅵其他银行卡吗？",
  "textb": "分期的如何用别的银行卡还钱",
  "label": 1,
  "id": 0
  }
```

标签映射：模型输出 0 表示不相似，输出 1 表示相似

```
"id2label":{
    "0":"not similarity",
    "1":"similarity"
     }
```

1.17.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.18 Erlangshen-Ubert-110M-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.18.1 简介 Brief Introduction

采用统一的框架处理多种抽取任务，AIWIN2022 的冠军方案，1.1 亿参数量的中文 UBERT-Base。

Adopting a unified framework to handle multiple information extraction tasks, AIWIN2022's champion solution, Chinese UBERT-Base (110M).

1.18.2 模型分类 Model Taxonomy

1.18.3 模型信息 Model Information

参考论文：Unified BERT for Few-shot Natural Language Understanding

UBERT 是2022 年 AIWIN 世界人工智能创新大赛：中文保险小样本多任务竞赛的冠军解决方案。我们开发了一个基于类似 BERT 的骨干的多任务、多目标、统一的抽取任务框架。我们的 UBERT 在比赛 A 榜和 B 榜上均取得了第一名。因为比赛中的数据集在比赛结束后不再可用，我们开源的 UBERT 从多个任务中收集了 70 多个数据集（共 1,065,069 个样本）来进行预训练，并且我们选择了MacBERT-Base作为骨干网络。除了支持开箱即用之外，我们的 UBERT 还可以用于各种场景，如 NLI、实体识别和阅读理解。示例代码可以在Github中找到。

UBERT was the winner solution in the 2022 AIWIN ARTIFICIAL INTELLIGENCE WORLD INNOVATIONS: Chinese Insurance Small Sample Multi-Task. We developed a unified framework based on BERT-like backbone for multiple tasks and objectives. Our UBERT owns first place, as described in leaderboards A and B. In addition to the unavailable datasets in the challenge, we carefully collect over 70 datasets (1,065,069 samples in total) from a variety of tasks for open-source UBERT. Moreover, we apply MacBERT-Base as the backbone. Besides out-of-the-box functionality, our UBERT can be employed in various scenarios such as NLI, entity recognition, and reading comprehension. The example codes can be found in Github.

- 论文：Unified BERT for Few-shot Natural Language Understanding

- 知乎：AIWIN 大赛冠军，IDEA 研究院封神榜提出多任务学习方案 Ubert

1.18.4 使用 Usage

模型下载地址 Download Address

加载模型 Loading Models

安装我们的 fengshen 框架，我们暂且提供如下方式安装

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
cd Fengshenbang-LM
pip install --editable ./
```

一键运行下面代码得到预测结果, 你可以任意修改示例 text 和要抽取的 entity_type，体验一下 Zero-Shot 性能

```
import argparse
from fengshen import UbertPiplines

total_parser = argparse.ArgumentParser("TASK NAME")
total_parser = UbertPiplines.piplines_args(total_parser)
args = total_parser.parse_args()
args.pretrained_model_path = 'IDEA-CCNL/Erlangshen-Ubert-110M-Chinese'
↪#预训练模型路径
test_data=[
    {
        "task_type": "抽取任务",
        "subtask_type": "实体识别",
        "text": "这也让很多业主据此认为，雅清苑是政府公务员挤对了国家的经适房政策。",
        "choices": [
            {"entity_type": "小区名字"},
            {"entity_type": "岗位职责"}
            ],
        "id": 0}
]

model = UbertPiplines(args)
result = model.predict(test_data)
for line in result:
    print(line)
```

Finetune 使用

开源的模型我们已经经过大量的数据进行预训练而得到，可以直接进行 Zero-Shot，如果你还想继续 finetune，可以参考我们的 example.py。你只需要将我们数据预处理成为我们定义的格式，即可使用简单的几行代码完成模型的训练和推理。我们是复用 pytorch-lightning 的 trainer 。在训练时，可以直接传入 trainer 的参数，此外我们还定义了一些其他参数。常用的参数如下：

```
--pretrained_model_path        #预训练模型的路径，默认
--load_checkpoints_path
↪#加载模型的路径，如果你finetune完，想加载模型进行预测可以传入这个参数
--batchsize                    #批次大小，默认 8
--monitor                      #保存模型需要监控的变量，例如我们可监控 val_span_acc
--checkpoint_path              #模型保存的路径，默认 ./checkpoint
--save_top_k                   #最多保存几个模型，默认 3
--every_n_train_steps          #多少步保存一次模型，默认 100
--learning_rate                #学习率，默认 2e-5
--warmup                       #预热的概率，默认 0.01
--default_root_dir             #模型日子默认输出路径
--gradient_clip_val            #梯度截断， 默认 0.25
--gpus                         #gpu 的数量
--check_val_every_n_epoch      #多少次验证一次， 默认 100
--max_epochs                   #多少个 epochs， 默认 5
--max_length                   #句子最大长度， 默认 512
--num_labels
↪#训练每条样本最多取多少个label，超过则进行随机采样负样本， 默认 10
```

数据预处理示例

整个模型的 Piplines 我们已经写好，所以为了方便，我们定义了数据格式。目前我们在预训练中主要含有一下几种任务类型

分类任务

普通分类任务

对于分类任务，我们把类别描述当作是 entity_type，我们主要关注 label 字段，label 为 1 表示该该标签是正确的标签。如下面示例所示

```
{
        "task_type": "分类任务",
        "subtask_type": "文本分类",
        "text": "7000亿美元救市方案将成期市毒药",
        "choices": [{
```

(续下页)

(接上页)

```
                "entity_type": "一则股票新闻",
                "label": 1,
                "entity_list": []
        }, {
                "entity_type": "一则教育新闻",
                "label": 0,
                "entity_list": []
        }, {
                "entity_type": "一则科学新闻",
                "label": 0,
                "entity_list": []
        }],
        "id": 0
}
```

自然语言推理

```
{
        "task_type": "分类任务",
        "subtask_type": "自然语言推理",
        "text": "在白云的蓝天下，一个孩子伸手摸着停在草地上的一架飞机的螺旋桨。",
        "choices": [{
                "entity_type": "可以推断出：一个孩子正伸手摸飞机的螺旋桨。",
                "label": 1,
                "entity_list": []
        }, {
                "entity_type": "不能推断出：一个孩子正伸手摸飞机的螺旋桨。",
                "label": 0,
                "entity_list": []
        }, {
                "entity_type": "很难推断出：一个孩子正伸手摸飞机的螺旋桨。",
                "label": 0,
                "entity_list": []
        }],
        "id": 0
}
```

语义匹配

```
{
        "task_type": "分类任务",
        "subtask_type": "语义匹配",
        "text": "不要借了我是试试看能否操作的",
        "choices": [{
                "entity_type": "不能理解为：借款审核期间能否取消借款",
                "label": 1,
                "entity_list": []
        }, {
                "entity_type": "可以理解为：借款审核期间能否取消借款",
                "label": 0,
                "entity_list": []
        }],
        "id": 0
}
```

抽取任务

对于抽取任务，label 字段是无效的

实体识别

```
{
        "task_type": "抽取任务",
        "subtask_type": "实体识别",
        "text":
↪"彭小军认为，国内银行现在走的是台湾的发卡模式，先通过跑马圈地再在圈的地里面选择客户，
↪",
        "choices": [{
                "entity_type": "地址",
                "label": 0,
                "entity_list": [{
                        "entity_name": "台湾",
                        "entity_type": "地址",
                        "entity_idx": [
                                [15, 16]
                        ]
                }]
        }{
                "entity_type": "政府机构",
```

(续下页)

(接上页)

```
                "label": 0,
                "entity_list": []
        }, {
                "entity_type": "电影名称",
                "label": 0,
                "entity_list": []
        }, {
                "entity_type": "人物姓名",
                "label": 0,
                "entity_list": [{
                        "entity_name": "彭小军",
                        "entity_type": "人物姓名",
                        "entity_idx": [
                                [0, 2]
                        ]
                }]
        },
        "id": 0
}
```

事件抽取

```
{
        "task_type": "抽取任务",
        "subtask_type": "事件抽取",
        "text": "小米9价格首降，6GB+128GB跌了200，却不如红米新机值得买",
        "choices": [{
                "entity_type": "降价的时间",
                "label": 0,
                "entity_list": []
        }, {
                "entity_type": "降价的降价方",
                "label": 0,
                "entity_list": []
        }, {
                "entity_type": "降价的降价物",
                "label": 0,
                "entity_list": [{
                        "entity_name": "小米9",
                        "entity_type": "降价的降价物",
                        "entity_idx": [
                                [0, 2]
```

(续下页)

(接上页)

```
                        ]
                }, {
                        "entity_name": "小米9",
                        "entity_type": "降价的降价物",
                        "entity_idx": [
                                [0, 2]
                        ]
                }]
        }, {
                "entity_type": "降价的降价幅度",
                "label": 0,
                "entity_list": []
        }],
        "id": 0
}
```

抽取式阅读理解

```
{
        "task_type": "抽取任务",
        "subtask_type": "抽取式阅读理解",
        "text":
↪"截至2014年7月1日，圣地亚哥人口估计为1381069人，是美国第八大城市，加利福尼亚州第二大城市。它是圣迭
↪蒂华纳城市群的一部分，是美国与底特律-
↪温莎之后的第二大跨境城市群，人口4922723。圣地亚哥是加州的出生地，以全年温和的气候、天然的深水港、广
↪",
        "choices": [{
                "entity_type": "除了医疗保健，圣迭戈哪个就业部门已经强势崛起？",
                "label": 0,
                "entity_list": [{
                        "entity_name": "生物技术发展",
                        "entity_idx": [
                                [153, 158]
                        ]
                }]
        }, {
                "entity_type": "在所有的军事部门中，哪一个在圣地亚哥的存在最为强大？",
                "label": 0,
                "entity_list": [{
                        "entity_name": "美国海军",
                        "entity_idx": [
                                [135, 138]
```

(续下页)

(接上页)

```
                        ]
                }]
        }, {
                "entity_type": "在美国十大城市中，圣迭戈排名哪一位？",
                "label": 0,
                "entity_list": [{
                        "entity_name": "第八",
                        "entity_idx": [
                                [33, 34]
                        ]
                }]
        }],
        "id": 0
}
```

1.18.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的对该模型的论文：

If you are using the resource for your work, please cite the our paper for this model:

```
@article{fengshenbang/ubert,
  author    = {JunYu Lu and
               Ping Yang and
               Jiaxing Zhang and
               Ruyi Gan and
               Jing Yang},
  title     = {Unified {BERT} for Few-shot Natural Language Understanding},
  journal   = {CoRR},
  volume    = {abs/2206.12094},
  year      = {2022}
}
```

如果您在您的工作中使用了我们的模型，也可以引用我们的总论文：

If you are using the resource for your work, please cite the our overview paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
```

(续下页)

(接上页)

```
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.19 Erlangshen-Ubert-330M-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.19.1 简介 Brief Introduction

采用统一的框架处理多种抽取任务，AIWIN2022 的冠军方案，3.3 亿参数量的中文 UBERT-Large。

Adopting a unified framework to handle multiple information extraction tasks, AIWIN2022's champion solution, Chinese UBERT-Large (330M).

1.19.2 模型分类 Model Taxonomy

1.19.3 模型信息 Model Information

参考论文：Unified BERT for Few-shot Natural Language Understanding

UBERT 是2022 年 AIWIN 世界人工智能创新大赛：中文保险小样本多任务竞赛的冠军解决方案。我们开发了一个基于类似 BERT 的骨干的多任务、多目标、统一的抽取任务框架。我们的 UBERT 在比赛 A 榜和 B 榜上均取得了第一名。因为比赛中的数据集在比赛结束后不再可用，我们开源的 UBERT 从多个任务中收集了 70 多个数据集（共 1,065,069 个样本）来进行预训练，并且我们选择了MacBERT-Large作为骨干网络。除了支持开箱即用之外，我们的 UBERT 还可以用于各种场景，如 NLI、实体识别和阅读理解。示例代码可以在Github中找到。

UBERT was the winner solution in the 2022 AIWIN ARTIFICIAL INTELLIGENCE WORLD INNOVATIONS: Chinese Insurance Small Sample Multi-Task. We developed a unified framework based on BERT-like backbone for multiple tasks and objectives. Our UBERT owns first place, as described in leaderboards A and B. In addition to the unavailable datasets in the challenge, we carefully collect over 70 datasets (1,065,069 samples in total) from a variety of tasks for open-source UBERT. Moreover, we apply MacBERT-Large as the backbone. Besides out-of-the-box functionality, our UBERT can be employed in various scenarios such as NLI, entity recognition, and reading comprehension. The example codes can be found in Github.

- 论文：Unified BERT for Few-shot Natural Language Understanding
- 知乎：AIWIN 大赛冠军，IDEA 研究院封神榜提出多任务学习方案 Ubert

1.19.4 使用 Usage

模型下载地址 Download Address

| 模型 | 地址 | |:---------:|:--------------:| | Erlangshen-Ubert-110M-Chinese | https://huggingface.co/IDEA-CCNL/Erlangshen-Ubert-110M-Chinese | | Erlangshen-Ubert-330M-Chinese | https://huggingface.co/IDEA-CCNL/Erlangshen-Ubert-330M-Chinese |

加载模型 Loading Models

Pip install fengshen

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
cd Fengshenbang-LM
pip install --editable ./
```

Run the code

```
import argparse
from fengshen import UbertPiplines

total_parser = argparse.ArgumentParser("TASK NAME")
total_parser = UbertPiplines.piplines_args(total_parser)
args = total_parser.parse_args()

args.pretrained_model_path = "IDEA-CCNL/Erlangshen-Ubert-330M-Chinese"

test_data=[
    {
        "task_type": "抽取任务",
        "subtask_type": "实体识别",
        "text": "这也让很多业主据此认为，雅清苑是政府公务员挤对了国家的经适房政策。",
```

(续下页)

(接上页)

```
        "choices": [
            {"entity_type": "小区名字"},
            {"entity_type": "岗位职责"}
            ],
        "id": 0}
]

model = UbertPiplines(args)
result = model.predict(test_data)
for line in result:
    print(line)
```

1.19.5 Finetune 使用

开源的模型我们已经经过大量的数据进行预训练而得到，可以直接进行 Zero-Shot，如果你还想继续 finetune，可以参考我们的 example.py。你只需要将我们数据预处理成为我们定义的格式，即可使用简单的几行代码完成模型的训练和推理。我们是复用 pytorch-lightning 的 trainer 。在训练时，可以直接传入 trainer 的参数，此外我们还定义了一些其他参数。常用的参数如下：

```
--pretrained_model_path        #预训练模型的路径，默认
--load_checkpoints_path
↪#加载模型的路径，如果你finetune完，想加载模型进行预测可以传入这个参数
--batchsize                    #批次大小，默认 8
--monitor                      #保存模型需要监控的变量，例如我们可监控 val_span_acc
--checkpoint_path              #模型保存的路径，默认 ./checkpoint
--save_top_k                   #最多保存几个模型，默认 3
--every_n_train_steps          #多少步保存一次模型，默认 100
--learning_rate                #学习率，默认 2e-5
--warmup                       #预热的概率，默认 0.01
--default_root_dir             #模型日子默认输出路径
--gradient_clip_val            #梯度截断，  默认 0.25
--gpus                         #gpu 的数量
--check_val_every_n_epoch      #多少次验证一次，  默认 100
--max_epochs                   #多少个 epochs，  默认 5
--max_length                   #句子最大长度，  默认 512
--num_labels
↪#训练每条样本最多取多少个label，超过则进行随机采样负样本，  默认 10
```

1.19.6 数据预处理示例

整个模型的 Piplines 我们已经写好，所以为了方便，我们定义了数据格式。目前我们在预训练中主要含有一下几种任务类型

分类任务

普通分类任务

对于分类任务，我们把类别描述当作是 entity_type，我们主要关注 label 字段，label 为 1 表示该该标签是正确的标签。如下面示例所示

```
{
        "task_type": "分类任务",
        "subtask_type": "文本分类",
        "text": "7000亿美元救市方案将成期市毒药",
        "choices": [{
                "entity_type": "一则股票新闻",
                "label": 1,
                "entity_list": []
        }, {
                "entity_type": "一则教育新闻",
                "label": 0,
                "entity_list": []
        }, {
                "entity_type": "一则科学新闻",
                "label": 0,
                "entity_list": []
        }],
        "id": 0
}
```

自然语言推理

```
{
        "task_type": "分类任务",
        "subtask_type": "自然语言推理",
        "text": "在白云的蓝天下，一个孩子伸手摸着停在草地上的一架飞机的螺旋桨。",
        "choices": [{
                "entity_type": "可以推断出：一个孩子正伸手摸飞机的螺旋桨。",
                "label": 1,
                "entity_list": []
```

(续下页)

(接上页)

```
        }, {
                "entity_type": "不能推断出：一个孩子正伸手摸飞机的螺旋桨。",
                "label": 0,
                "entity_list": []
        }, {
                "entity_type": "很难推断出：一个孩子正伸手摸飞机的螺旋桨。",
                "label": 0,
                "entity_list": []
        }],
        "id": 0
}
```

语义匹配

```
{
        "task_type": "分类任务",
        "subtask_type": "语义匹配",
        "text": "不要借了我是试试看能否操作的",
        "choices": [{
                "entity_type": "不能理解为：借款审核期间能否取消借款",
                "label": 1,
                "entity_list": []
        }, {
                "entity_type": "可以理解为：借款审核期间能否取消借款",
                "label": 0,
                "entity_list": []
        }],
        "id": 0
}
```

抽取任务

对于抽取任务，label 字段是无效的

实体识别

```
{
        "task_type": "抽取任务",
        "subtask_type": "实体识别",
        "text":
↪"彭小军认为，国内银行现在走的是台湾的发卡模式，先通过跑马圈地再在圈的地里面选择客户，
↪",
        "choices": [{
                "entity_type": "地址",
                "label": 0,
                "entity_list": [{
                        "entity_name": "台湾",
                        "entity_type": "地址",
                        "entity_idx": [
                                [15, 16]
                        ]
                }]
        }{
                "entity_type": "政府机构",
                "label": 0,
                "entity_list": []
        }, {
                "entity_type": "电影名称",
                "label": 0,
                "entity_list": []
        }, {
                "entity_type": "人物姓名",
                "label": 0,
                "entity_list": [{
                        "entity_name": "彭小军",
                        "entity_type": "人物姓名",
                        "entity_idx": [
                                [0, 2]
                        ]
                }]
        },
        "id": 0
}
```

事件抽取


```
{
        "task_type": "抽取任务",
        "subtask_type": "事件抽取",
        "text": "小米9价格首降，6GB+128GB跌了200，却不如红米新机值得买",
        "choices": [{
                "entity_type": "降价的时间",
                "label": 0,
                "entity_list": []
        }, {
                "entity_type": "降价的降价方",
                "label": 0,
                "entity_list": []
        }, {
                "entity_type": "降价的降价物",
                "label": 0,
                "entity_list": [{
                        "entity_name": "小米9",
                        "entity_type": "降价的降价物",
                        "entity_idx": [
                                [0, 2]
                        ]
                }, {
                        "entity_name": "小米9",
                        "entity_type": "降价的降价物",
                        "entity_idx": [
                                [0, 2]
                        ]
                }]
        }, {
                "entity_type": "降价的降价幅度",
                "label": 0,
                "entity_list": []
        }],
        "id": 0
}
```

抽取式阅读理解

```
{
        "task_type": "抽取任务",
        "subtask_type": "抽取式阅读理解",
        "text":
↪"截至2014年7月1日，圣地亚哥人口估计为1381069人，是美国第八大城市，加利福尼亚州第二大城市。它是圣迭
↪蒂华纳城市群的一部分，是美国与底特律-
↪温莎之后的第二大跨境城市群，人口4922723。圣地亚哥是加州的出生地，以全年温和的气候、天然的深水港、广
↪",
        "choices": [{
                "entity_type": "除了医疗保健，圣迭戈哪个就业部门已经强势崛起？",
                "label": 0,
                "entity_list": [{
                        "entity_name": "生物技术发展",
                        "entity_idx": [
                                [153, 158]
                        ]
                }]
        }, {
                "entity_type": "在所有的军事部门中，哪一个在圣地亚哥的存在最为强大？",
                "label": 0,
                "entity_list": [{
                        "entity_name": "美国海军",
                        "entity_idx": [
                                [135, 138]
                        ]
                }]
        }, {
                "entity_type": "在美国十大城市中，圣迭戈排名哪一位？",
                "label": 0,
                "entity_list": [{
                        "entity_name": "第八",
                        "entity_idx": [
                                [33, 34]
                        ]
                }]
        }],
        "id": 0
}
```

1.19.7 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的对该模型的论文：

If you are using the resource for your work, please cite the our paper for this model:

```
@article{fengshenbang/ubert,
  author    = {JunYu Lu and
               Ping Yang and
               Jiaxing Zhang and
               Ruyi Gan and
               Jing Yang},
  title     = {Unified {BERT} for Few-shot Natural Language Understanding},
  journal   = {CoRR},
  volume    = {abs/2206.12094},
  year      = {2022}
}
```

如果您在您的工作中使用了我们的模型，也可以引用我们的总论文：

If you are using the resource for your work, please cite the our overview paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.20 Erlangshen-ZEN1-224M-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.20.1 简介 Brief Introduction

善于处理 NLU 任务，使用了 N-gram 编码增强文本语义，2.24 亿参数量的 ZEN1

ZEN1 model, which uses N-gram to enhance text semantic and has 224M parameters, is adept at NLU tasks.

1.20.2 模型分类 Model Taxonomy

1.20.3 模型信息 Model Information

我们与ZEN 团队合作，使用我们的封神框架，开源发布了 ZEN1 模型。具体而言，通过引入无监督学习中提取的知识，ZEN 通过 N-gram 方法学习不同的文本粒度信息。ZEN1 可以通过仅在单个小语料库 (低资源场景) 上进行训练来获得良好的性能增益。下一步，我们将继续与 ZEN 团队一起探索 PLM 的优化，并提高下游任务的性能

We open source and publicly release ZEN1 using our Fengshen Framework in collaboration with the ZEN team. More precisely, by bringing together knowledge extracted by unsupervised learning, ZEN learns different textual granularity information through N-gram methods. ZEN1 can obtain good performance gains by training only on a single small corpus (low-resource scenarios). In the next step, we continue with the ZEN team to explore the optimization of PLM and improve the performance on downstream tasks.

下游效果 Performance

分类任务 Classification

抽取任务 Extraction

1.20.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-ZEN1-224M-Chinese

加载模型 Loading Models

因为transformers库中是没有 ZEN1 相关的模型结构的，所以你可以在我们的Fengshenbang-LM中找到并且运行代码。

Since there is no structure of ZEN1 in transformers library, you can find the structure of ZEN1 and run the codes in Fengshenbang-LM.

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
```

```
from fengshen.models.zen1.ngram_utils import ZenNgramDict
from fengshen.models.zen1.tokenization import BertTokenizer
from fengshen.models.zen1.modeling import ZenForSequenceClassification,␣
↪ZenForTokenClassification

pretrain_path = 'IDEA-CCNL/Erlangshen-ZEN1-224M-Chinese'

tokenizer = BertTokenizer.from_pretrained(pretrain_path)
model_classification = ZenForSequenceClassification.from_pretrained(pretrain_path)
model_extraction = ZenForTokenClassification.from_pretrained(pretrain_path)
ngram_dict = ZenNgramDict.from_pretrained(pretrain_path, tokenizer=tokenizer)
```

你可以从下方的链接获得我们做分类和抽取的详细示例。

You can get classification and extraction examples below.

分类 classification example on fengshen

抽取 extraction example on fengshen

1.20.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的对该模型的论文：

If you are using the resource for your work, please cite the our paper for this model:

```
@inproceedings{fengshenbang/zen1,
  author    = {Shizhe Diao and
               Jiaxin Bai and
               Yan Song and
               Tong Zhang and
               Yonggang Wang},
  title     = {{ZEN:} Pre-training Chinese Text Encoder Enhanced by N-gram␣
↪Representations},
  booktitle = {{EMNLP} (Findings)},
  series    = {Findings of {ACL}},
```

(续下页)

(接上页)

```
  volume    = {{EMNLP} 2020},
  pages     = {4729--4740},
  publisher = {Association for Computational Linguistics},
  year      = {2020}
}
```

如果您在您的工作中使用了我们的模型，也可以引用我们的总论文:

If you are using the resource for your work, please cite the our overview paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

1.21 Erlangshen-ZEN2-668M-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

1.21.1 简介 Brief Introduction

善于处理 NLU 任务，使用了 N-gram 编码增强文本语义，6.68 亿参数量的 ZEN2

ZEN2 model, which uses N-gram to enhance text semantic and has 668M parameters, is adept at NLU tasks.

1.21.2 模型分类 Model Taxonomy

1.21.3 模型信息 Model Information

我们与ZEN 团队合作，使用我们的封神框架，开源发布了 ZEN2 模型。具体而言，通过引入无监督学习中提取的知识，ZEN 通过 N-gram 方法学习不同的文本粒度信息。ZEN2 使用大规模数据集和特殊的预训练策略对 N-gram 增强编码器进行预训练。下一步，我们将继续与 ZEN 团队一起探索 PLM 的优化，并提高下游任务的性能。

We open source and publicly release ZEN2 using our Fengshen Framework in collaboration with the ZEN team. More precisely, by bringing together knowledge extracted by unsupervised learning, ZEN learns different textual granularity information through N-gram methods. ZEN2 pre-trains the N-gram-enhanced encoders with large-scale datasets and special pre-training strategies. In the next step, we continue with the ZEN team to explore the optimization of PLM and improve the performance on downstream tasks.

下游效果 Performance

分类任务 Classification

抽取任务 Extraction

1.21.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Erlangshen-ZEN2-668M-Chinese

加载模型 Loading Models

因为transformers库中是没有 ZEN2 相关的模型结构的，所以你可以在我们的Fengshenbang-LM中找到并且运行代码。

Since there is no structure of ZEN2 in transformers library, you can find the structure of ZEN2 and run the codes in Fengshenbang-LM.

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
```

```
from fengshen.models.zen2.ngram_utils import ZenNgramDict
from fengshen.models.zen2.tokenization import BertTokenizer
from fengshen.models.zen2.modeling import ZenForSequenceClassification,␣
↪ZenForTokenClassification

pretrain_path = 'IDEA-CCNL/Erlangshen-ZEN2-668M-Chinese'

tokenizer = BertTokenizer.from_pretrained(pretrain_path)
model_classification = ZenForSequenceClassification.from_pretrained(pretrain_path)
model_extraction = ZenForTokenClassification.from_pretrained(pretrain_path)
ngram_dict = ZenNgramDict.from_pretrained(pretrain_path, tokenizer=tokenizer)
```

你可以从下方的链接获得我们做分类和抽取的详细示例。

You can get classification and extraction examples below.

分类 classification example on fengshen

抽取 extraction example on fengshen

1.21.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的对该模型的论文：

If you are using the resource for your work, please cite the our paper for this model:

```
@article{Sinovation2021ZEN2,
  title="{ZEN 2.0: Continue Training and Adaption for N-gram Enhanced Text Encoders}",
  author={Yan Song, Tong Zhang, Yonggang Wang, Kai-Fu Lee},
  journal={arXiv preprint arXiv:2105.01279},
  year={2021},
}
```

如果您在您的工作中使用了我们的模型，也可以引用我们的总论文：

If you are using the resource for your work, please cite the our overview paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
```

(续下页)

(接上页)

```
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

CHAPTER 2

闻仲系列

2.1 Wenzhong-GPT2-3.5B

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

2.1.1 简介 Brief Introduction

善于处理 NLG 任务，目前最大的，中文版的 GPT2

Focused on handling NLG tasks, the current largest, Chinese GPT2.

2.1.2 模型分类 Model Taxonomy

2.1.3 模型信息 Model Information

为了可以获得一个强大的单向语言模型，我们采用 GPT 模型结构，并且应用于中文语料上。具体地，这个模型拥有 30 层解码器和 35 亿参数，这比原本的 GPT2-XL 还要大。我们在 100G 的中文语料上预训练，这消耗了 32 个 NVIDIA A100 显卡大约 28 小时。据我们所知，它是目前最大的中文的 GPT 模型。

To obtain a robust unidirectional language model, we adopt the GPT model structure and apply it to the Chinese corpus. Specifically, this model has 30 decoder layers and 3.5 billion parameters, which is larger than the original GPT2-XL. We pre-train it on 100G of Chinese corpus, which consumes 32 NVIDIA A100 GPUs for about 28 hours. To the best of our knowledge, it is the largest Chinese GPT model currently available.

2.1.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Wenzhong-GPT2-3.5B

加载模型 Loading Models

```
from transformers import GPT2Tokenizer, GPT2Model
tokenizer = GPT2Tokenizer.from_pretrained('IDEA-CCNL/Wenzhong-GPT2-3.5B')
model = GPT2Model.from_pretrained('IDEA-CCNL/Wenzhong-GPT2-3.5B')
text = "Replace me by any text you'd like."
encoded_input = tokenizer(text, return_tensors='pt')
output = model(**encoded_input)
```

使用示例 Usage Examples

```
from transformers import pipeline, set_seed
set_seed(55)
generator = pipeline('text-generation', model='IDEA-CCNL/Wenzhong-GPT2-3.5B')
generator("北京位于", max_length=30, num_return_sequences=1)
```

2.1.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

2.2 Wenzhong-GPT2-110M

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

2.2.1 简介 Brief Introduction

善于处理 NLG 任务，中文版的 GPT2-Small。

Focused on handling NLG tasks, Chinese GPT2-Small.

2.2.2 模型分类 Model Taxonomy

2.2.3 模型信息 Model Information

类似于 Wenzhong2.0-GPT2-3.5B-chinese，我们实现了一个 small 版本的 12 层的 Wenzhong-GPT2-110M，并且在悟道（300G 版本）上面进行预训练。

Similar to Wenzhong2.0-GPT2-3.5B-chinese, we implement a small size Wenzhong-GPT2-110M with 12 layers, which is pre-trained on Wudao Corpus (300G version).

2.2.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Wenzhong-GPT2-110M

加载模型 Loading Models

```
from transformers import GPT2Tokenizer,GPT2LMHeadModel
hf_model_path = 'IDEA-CCNL/Wenzhong-GPT2-110M'
tokenizer = GPT2Tokenizer.from_pretrained(hf_model_path)
model = GPT2LMHeadModel.from_pretrained(hf_model_path)
```

使用示例 Usage Examples

```
question = "北京是中国的"
inputs = tokenizer(question,return_tensors='pt')
generation_output = model.generate(**inputs,
                                   return_dict_in_generate=True,
                                   output_scores=True,
                                   max_length=150,
                                   # max_new_tokens=80,
                                   do_sample=True,
                                   top_p = 0.6,
                                   # num_beams=5,
                                   eos_token_id=50256,
                                   pad_token_id=0,
                                   num_return_sequences = 5)

for idx,sentence in enumerate(generation_output.sequences):
    print('next sentence %d:\n'%idx,
    tokenizer.decode(sentence).split('<|endoftext|>')[0])
    print('*'*40)
```

2.2.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
```

(续下页)

(接上页)

```
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

2.3 Wenzhong2.0-GPT2-3.5B-chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

2.3.1 简介 Brief Introduction

基于悟道数据集预训练，善于处理 NLG 任务，目前最大的，中文版的 GPT2。

Pretraining on Wudao Corpus, focused on handling NLG tasks, the current largest, Chinese GPT2.

2.3.2 模型分类 Model Taxonomy

2.3.3 模型信息 Model Information

为了可以获得一个强大的单向语言模型，我们采用 GPT 模型结构，并且应用于中文语料上。类似于 Wenzhong-GPT2-3.5B，这个模型拥有 30 层解码器和 35 亿参数，这比原本的 GPT2-XL 还要大。不同的是，我们把这个模型在悟道（300G 版本）语料上进行预训练。据我们所知，它是目前最大的中文的 GPT 模型。

To obtain a powerful unidirectional language model, we adopt the GPT model structure and apply it to the Chinese corpus. Similar to Wenzhong-GPT2-3.5B, this model has 30 decoder layers and 3.5 billion parameters, which is larger than the original GPT2-XL. The difference is that we pre-trained this model on the Wudao (300G version) corpus. To the best of our knowledge, it is the largest Chinese GPT model currently available.

2.3.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Wenzhong2.0-GPT2-3.5B-chinese

加载模型 Loading Models

```
from transformers import GPT2Tokenizer, GPT2Model
tokenizer = GPT2Tokenizer.from_pretrained('IDEA-CCNL/Wenzhong2.0-GPT2-3.5B-chinese')
model = GPT2Model.from_pretrained('IDEA-CCNL/Wenzhong2.0-GPT2-3.5B-chinese')
text = "Replace me by any text you'd like."
encoded_input = tokenizer(text, return_tensors='pt')
output = model(**encoded_input)
```

使用示例 Usage Examples

```
from transformers import pipeline, set_seed
set_seed(55)
generator = pipeline('text-generation', model='IDEA-CCNL/Wenzhong2.0-GPT2-3.5B-chinese
↪')
generator("北京位于", max_length=30, num_return_sequences=1)
```

2.3.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

CHAPTER 3

燃灯系列

3.1 Randeng-BART-139M-SUMMARY

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.1.1 简介 Brief Introduction

善于处理摘要任务，在一个中文摘要数据集上微调后的，中文版的 BART-base。

Good at solving text summarization tasks, after fine-tuning on a Chinese text summarization dataset, Chinese BART-base.

3.1.2 模型分类 Model Taxonomy

3.1.3 模型信息 Model Information

基于Randeng-BART-139M，我们在收集的 1 个中文领域的文本摘要数据集（LCSTS）上微调了它，得到了 summary 版本。

Based on 基于Randeng-BART-139M, we fine-tuned a text summarization version (summary) on a Chinese text summarization datasets (LCSTS).

3.1.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Randeng-BART-139M-SUMMARY

加载模型 Loading Models

```
from transformers import BartForConditionalGeneration, AutoTokenizer,␣
↪Text2TextGenerationPipeline
import torch

tokenizer=AutoTokenizer.from_pretrained('IDEA-CCNL/Randeng-BART-139M-SUMMARY')
model=BartForConditionalGeneration.from_pretrained('IDEA-CCNL/Randeng-BART-139M-
↪SUMMARY')
text =
↪'summary:在北京冬奥会自由式滑雪女子坡面障碍技巧决赛中，中国选手谷爱凌夺得银牌。祝贺谷爱凌！今天上午
↪90分。在12位选手中排名第三。完成动作后，谷爱凌又扮了个鬼脸，甚是可爱。第二轮中，谷爱凌在道具区第三
↪98分。网友：摔倒了也没关系，继续加油！在第二跳失误摔倒的情况下，谷爱凌顶住压力，第三跳稳稳发挥，流
↪23分！此轮比赛，共12位选手参赛，谷爱凌第10位出场。网友：看比赛时我比谷爱凌紧张，加油！
↪'
text2text_generator = Text2TextGenerationPipeline(model, tokenizer)
print(text2text_generator(text, max_length=50, do_sample=False))
```

3.1.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.2 Randeng-BART-139M

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.2.1 简介 Brief Introduction

善于处理 NLT 任务，中文版的 BART-base。

Good at solving NLT tasks, Chinese BART-base.

3.2.2 模型分类 Model Taxonomy

3.2.3 模型信息 Model Information

参考论文：BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

为了得到一个中文版的 BART-base，我们用悟道语料库 (180G 版本) 进行预训练。具体地，我们在预训练阶段中使用了封神框架大概花费了 8 张 A100 约 3 天。

To get a Chinese BART-base, we use WuDao Corpora (180 GB version) for pre-training. Specifically, we use the fengshen framework in the pre-training phase which cost about 3 days with 8 A100 GPUs.

更多信息 More Information

BART 模型在原论文中采用了 5 种 Denoise 的方式，我们在预训练的时候采用论文中效果比较好的 text infilling 的方式。同时在原论文的基础上，我们使用 sentence piece tokenizer，使得模型具备更长文本的生成能力。

3.2.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Randeng-BART-139M

加载模型 Loading Models

因为 transformers 下 BartTokenizer 不支持 sentence piece，所以这里借用的是 T5Tokenizer，在使用时需要在句首手动添加 <s> (bos_token) ^ ^

```
from transformers import BartForConditionalGeneration, AutoTokenizer,␣
↪Text2TextGenerationPipeline
import torch

tokenizer=AutoTokenizer.from_pretrained('IDEA-CCNL/Randeng-BART-139M', use_fast=false)
model=BartForConditionalGeneration.from_pretrained('IDEA-CCNL/Randeng-BART-139M')
text = '<s>桂林市是世界闻名<mask> ，它有悠久的<mask>'
text2text_generator = Text2TextGenerationPipeline(model, tokenizer)
print(text2text_generator(text, max_length=50, do_sample=False))
```

3.2.5 如何预训练自己的 BART 模型

我们的训练代码、训练脚本都已开源，可以在fengshen/examples/pretrain_bart找到。

数据预处理

我们的原始数据并没有经过太多的数据处理，原始数据仅仅过简单的分句、分词。一个输入的 sample 如下：

```
[['_在', '网上', '找', '了很久', '都没有', '关于', 'j', 'ava', '的', '网页', '超',
↪'链接', '跳', '转', '方式', ',', '故', '写', '篇', '经验', '供', '大家', '分享', '。
↪'],
['_下面', '将', '介绍', '如何在', '文本', '消息', '中使用', '网页', '超', '链接', ':',
↪ '_其实', ',', '不知道', '如何在', '文本', '消息', '中使用', '网页', '超', '链接',
↪'的开发', '者', '几乎', '100%', '都', '熟悉', 'l', ',', '特别是', '对', 'l', '中的',
↪ 'a', '标签', '再', '熟悉', '不过', '了', '。'],
['_那', '到底', '在', '微信', '公众', '帐', '号的', '文本', '消息', '中使用', '超',
↪'链接', '要注意', ',', '在', '微信', '上', ',', 'l', '的', 'a', '标签', '属性', '值
↪', '不用', '引', '号', '引起', ',', '或者', '使用', '单', '引', '号', '引起', ',',
↪'都是', '错误的', '写', '法', '(', '在', 'iphone', '上', ',', 'a', '标签', '属性',
↪'h', 're', 'f', '的', '值', '用', '单', '引', '号', '是', '正常的', ')。'],
['_', '正确的', '用法', '是将', 'a', '标签', 'h', 're', 'f', '属', '性的', '值', '用',
```

(续下页)

(接上页)

```
↪ '双', '引', '号', '引起', ',', '代码', '如下', ':', '_a', '_h', 're', 'f', '="',
↪'_.', 'com', '"', '百度', '经验', 'a', '_如果', '要', '使用', '超', '链接', '调用',
↪'action', '类', ',', '可以在', '要', '输出的', 't', 'ext', '文本', '中', '拼接',
↪'如下', ':', '_', '一定要在', 'action', '前', '加入', '在', '微信', '发布的', '_r',
↪'l', '。'],
['_str', 'ing', '_m', 'sg', '="', '超', '链接', ':', 'a', '_h', 're', 'f', '=', '_工程
↪', '名', 'we', 'ix', 'in', '.', 'do', '?'],
['_act', 'ion', '=', 'xx', 'x', '&', 'a', '=', '2', '_', '跳', '转', 'a', '";']]
```

最终由上述数据转成模型输入的函数可以在 TextFillingCollator 中找到。

脚本修改

用户仅需要简单修改script 脚本，即可快速分布式的训练我们的 BART。

其中参数主要分成下面五个参数:

```
数据类参数，主要用于配制数据集Batch␣
↪Size等参数，在后续我们开源数据集工程后，能大幅减少数据预处理的工作量。

DATA_ARGS="\
        --datasets_name wudao_180g_spbpe_tokenized \
        --pretrain_sp_tokenizer /cognitive_comp/common_data/tokenizers/sentence_piece_
↪bpe/bpe_v40000_s42_cov0.9995_max6_corpus1M.model \
        --num_workers 30 \
        --train_batchsize $MICRO_BATCH_SIZE \
        --val_batchsize 32 \
        --test_batchsize 32  \
        --max_seq_length 1024 \
        --masked_lm_prob 0.15 \
        --val_datasets_field test \
        "
```

```
模型的参数会从model_path中自动获取，可以参考我们Huggingface的模型配置，在上面做修改。

learning_rate、weight_
↪decay这些参数如果配置了Deepspeed配置，会从Deepspeed的配置中获取。

MODEL_ARGS="\
        --model_path IDEA-CCNL/Randeng-BART-139M \
        --learning_rate 1e-5 \
        --weight_decay 0.1 \
        --warmup 0.001 \
        "
```

```
模型保存类参数，用户可以根据自己需要设定

MODEL_CHECKPOINT_ARGS="\
        --monitor train_loss \
        --save_top_k 3 \
        --mode min \
        --save_last \
        --every_n_train_steps 50000 \
        --dirpath /cognitive_comp/gaoxinyu/ln_model/ckpt/fengshen-$MODEL_NAME \
        --filename model-{step:02d}-{train_loss:.4f} \
```

```
训练相关的参数，这里可以根据自己的机器需要，调整gpus、nodes、strategy等等，如果使用Deepspeed，Deepspe
export PL_DEEPSPEED_CONFIG_PATH=$config_json

TRAINER_ARGS="\
        --gradient_clip_val 1.0 \
        --max_epochs 1 \
        --gpus 1 \
        --num_nodes 1 \
        --strategy deepspeed_stage_1 \
        --log_every_n_steps 100 \
        --val_check_interval 0.1 \
        --accumulate_grad_batches 1 \
        --resume_from_checkpoint /cognitive_comp/gaoxinyu/ln_model/ckpt/fengshen-$
↪{MODEL_NAME}/last.ckpt \
        --default_root_dir /cognitive_comp/gaoxinyu/ln_model/fengshen-$MODEL_NAME \
        "
```

3.2.6 如何进行下游任务

这里我们提供利用 BART 做 summary 任务的示例。代码脚本同样开源了。

在脚本中仅需要修改一下 LSCTC 数据的地址即可。

3.2.7 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
```

(续下页)

(接上页)

```
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.3 Randeng-BART-759M-Chinese-BertTokenizer

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.3.1 简介 Brief Introduction

善于处理 NLT 任务，使用 BERT 分词器，大规模的中文版的 BART。

Good at solving NLT tasks, applying the BERT tokenizer, a large-scale Chinese BART.

3.3.2 模型分类 Model Taxonomy

3.3.3 模型信息 Model Information

为了得到一个大规模的中文版的 BART(约 BART-large 的两倍)，我们用悟道语料库 (180G 版本) 进行预训练。具体地，我们在预训练阶段中使用了封神框架大概花费了 8 张 A100 约 7 天。值得注意的是，因为 BERT 分词器通常在中文任务中表现比其他分词器好，所以我们使用了它。我们也开放了我们预训练的代码：pre-train_randeng_bart。

To obtain a large-scale Chinese BART (around twice as large as BART-large), we use WuDao Corpora (180 GB version) for pre-training. Specifically, we use the fengshen framework in the pre-training phase which cost about 7 days with 8 A100 GPUs. Note that since the BERT tokenizer usually performs better than others for Chinese tasks, we employ it. We have also released our pre-training code: pretrain_randeng_bart.

3.3.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Randeng-BART-759M-Chinese-BertTokenizer

加载模型 Loading Models

```
from transformers import BartForConditionalGeneration, AutoTokenizer,␣
↪Text2TextGenerationPipeline
import torch

tokenizer=AutoTokenizer.from_pretrained('IDEA-CCNL/Randeng-BART-759M-Chinese-
↪BertTokenizer', use_fast=false)
model=BartForConditionalGeneration.from_pretrained('IDEA-CCNL/Randeng-BART-759M-
↪Chinese-BertTokenizer')
text = '桂林是著名的[MASK]，它有很多[MASK]。'
text2text_generator = Text2TextGenerationPipeline(model, tokenizer)
print(text2text_generator(text, max_length=50, do_sample=False))
```

3.3.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.4 Randeng-DAVAE-1.2B-General-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.4.1 简介 Brief Introduction

使用 101M 的 Bert 作为 encoder，1.1B 参数量的 transformer-XL 作为 decoder，以此构成变分自编码 (VAE) 网络。在训练中为了得到更好的潜在表示，对输入的 embedding 施加连续 gaussian noise 并且使用对抗学习训练后验网络，这就是 DAVAE 的由来。

The Variational Autoencoder (VAE) network comprises an encoder using Bert with 101M parameters and a decoder using transformer-XL with 1.1B parameters. To make the representation more expressive, the input embedding is perturbed with gaussian noise, and adversarial learning is used to train the posterior network, so forming the DAVAE.

3.4.2 模型分类 Model Taxonomy

3.4.3 模型信息 Model Information

数据准备 Corpus Preparation

- 悟道语料库（280G 版本）
- Wudao Corpus (with 280G samples)

防止后验崩塌 Avoiding posterior collapse

为了防止在通用语料上后验崩塌，我们在训练中加入以下措施，

1. 使用 KL annealing。对于正则项系数，采用梯形 scheduler
2. 加入 free bits。设置 free bits，避免过分靠近先验
3. 强化潜在向量引导。潜在空间向量和 decoder 隐层输出逐位相加

4. 在输入 embedding 上加入连续 gaussian 噪声，与之前工作使用离散加噪方式不同
5. 在潜在空间进行对抗训练

We used several methods to avoid posterior collapse, as what follows,

1. Using KL annealing. A trapezoidal scheduler was used to calculate the coefficient for the regularization term.
2. Adding free-bits constraint. we chose a certain free bit to avoid getting too close to the prior in the training.
3. Strengthening the guidance of the latent vector. The latent vector was added over the hidden state of every token.
4. Adding gaussian noise to the input embedding, differing from the noising method used in previous work.
5. Adversarial training in latent space.

3.4.4 使用 Usage

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
cd Fengshenbang-LM
pip install --editable .
```

```
import torch
from fengshen.models.DAVAE.DAVAEModel import DAVAEModel
from transformers import BertTokenizer,T5Tokenizer
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

encoder_tokenizer = BertTokenizer.from_pretrained("IDEA-CCNL/Randeng-DAVAE-1.2B-
↪General-Chinese")
decoder_tokenizer = T5Tokenizer.from_pretrained("IDEA-CCNL/Randeng-DAVAE-1.2B-General-
↪Chinese", eos_token = '<|endoftext|>', pad_token = '<pad>',extra_ids=0)
decoder_tokenizer.add_special_tokens({'bos_token':'<bos>'})
vae_model = DAVAEModel.from_pretrained("IDEA-CCNL/Randeng-DAVAE-1.2B-General-Chinese
↪").to(device)
input_texts = [
    "针对电力系统中的混沌振荡对整个互联电网的危害问题,
↪提出了一种基于非线性光滑函数的滑模控制方法.",
    "超市面积不算大.挺方便附近的居民购买的. 生活用品也比较齐全.价格适用中.",
]
output_texts = vae_model.simulate_batch(encoder_tokenizer,decoder_tokenizer,input_
↪texts)
print(output_texts)
```

3.4.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的网站:

If you are using the resource for your work, please cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.5 Randeng-DELLA-226M-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.5.1 简介 Brief Introduction

在悟道数据集上进行通用预训练的 Deep VAE 模型。其中编码器和解码器都是 GPT-2 架构。可以用于下游的句子重写，语义转换，性质控制等任务。

A deep VAE model pretrained on Wudao dataset. Both encoder and decoder are based on GPT-2 architecture. Such model is particularly suitable for paraphrasing, semantic updating and fine-grained attributes control.

3.5.2 模型分类 Model Taxonomy

3.5.3 模型信息 Model Information

参考论文 Reference Paper：Fuse It More Deeply! A Variational Transformer with Layer-Wise Latent Variable Inference for Text Generation

本模型使用了 Della 论文里的循环潜在向量架构，但对于解码器生成并未采用原论文的 low-rank-tensor-product 来进行信息融合，而是使用了简单的线性变换后逐位逐词添加的方式。该方式对于开放域数据集的预训练稳定性有较大正向作用。

Note that although we adopted the layer-wise recurrent latent variables structure as the paper, we did not use the low-rank-tensor-product to fuse the latent vectors to the decoder hidden states. Instead we applied a simple linear transformation on the latent vectors and then add them to the hidden states independently.

3.5.4 使用 Usage

```
# Checkout the latest Fengshenbang-LM directory and run following script under␣
↪Fengshenbang-LM root directory
import torch
from torch.nn.utils.rnn import pad_sequence
from fengshen.models.deepVAE.deep_vae import Della
from transformers.models.bert.tokenization_bert import BertTokenizer
tokenizer = BertTokenizer.from_pretrained("IDEA-CCNL/Randeng-DELLA-226M-Chinese")
vae_model = Della.from_pretrained("IDEA-CCNL/Randeng-DELLA-226M-Chinese")
special_tokens_dict = {'bos_token': '<BOS>', 'eos_token': '<EOS>'}
tokenizer.add_special_tokens(special_tokens_dict)
sentence =
↪"本模型是在通用数据集下预训练的VAE模型，如要获得最佳效果请在特定领域微调后使用。"
tokenized_text = tokenizer.convert_tokens_to_ids(tokenizer.tokenize(sentence))
decoder_target = [tokenizer.bos_token_id] + tokenized_text + [tokenizer.eos_token_id]
inputs = []
inputs.append(torch.tensor(decoder_target, dtype=torch.long))
inputs = pad_sequence(inputs, batch_first=True, padding_value=0)
max_length = 256
top_p = 0.5
top_k = 0
temperature = .7
repetition_penalty = 1.0
sample = False
device = 0
model = vae_model.eval()
model = model.to(device)
outputs = model.model.inference(inputs.to(device), top_p=top_p, top_k=top_k, max_
↪length=max_length, sample=sample,
    temperature=temperature, repetition_penalty=repetition_penalty)
for gen_sent, orig_sent in zip(outputs, inputs):
    print('orig_sent:', tokenizer.decode(orig_sent).replace(' ', ''))
    print('gen_sent:', tokenizer.decode(gen_sent).replace(' ', ''))
    print("-"*20)
```

3.5.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.6 Randeng-GAVAE-1.2B-Augmentation-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.6.1 简介 Brief Introduction

GAVAE(Generative Adversarial Varational Auto-encoder) 在预训练 VAE 模型的隐空间插入 GAN 网络，对类别文本的隐向量进行对抗生成训练，以少量特定类别文本训练后即可生成该类别文本。

GAVAE (Generative Adversarial Varational Auto-encoder) inserts a GAN model into the hidden space of the pre-trained VAE model and performs generative Adversarial training on the hidden vectors of a small number of categories of text, which can be used to generate that category of text after training.

3.6.2 模型分类 Model Taxonomy

3.6.3 模型信息 Model Information

Pretrained VAE:

训练语料：悟道语料库（280G 版本）

Training Corpus: Wudao Corpus (with 280G samples)

参考模型：Randeng-DAVAE-1.2B-General-Chinese

Reference model:Randeng-DAVAE-1.2B-General-Chinese

GAN

生成器：五层 MLP，生成类别隐向量；

判别器：三层 MLP，判断向量为真实类别隐向量或生成器生成的向量。

训练语料：少量类别文本。

Generator: five-layer MLP, generating category hidden vectors.

Discriminator: three-layer MLP, which determines whether the vector is a true category hidden vector or a generated vector.

Training corpus: a small amount of categorical text.

3.6.4 使用 Usage

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
cd Fengshenbang-LM
pip install --editable .
```

```
import torch
from transformers import BertTokenizer,T5Tokenizer
from fengshen.models.GAVAE.GAVAEModel import GAVAEModel
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
input_texts = [

↪"非常好的一个博物馆，是我所有去过的博物馆里感觉最正规的一家，凭有效证件可以入馆，可以自助免费存小件
↪",

↪"这是我来长沙最最期待的一定要去的地方，总算今天特地去瞻仰千古遗容了，开车到门口大屏幕显示着门票已发
↪",
    "地方很大 很气派~~可以逛很久~~~去的时候是免费的~不过要安检~~~里面的马王堆~
↪幸追夫人~还是很不错的~~~~去的时候有一个吴越文化特别展~~~东西也很多~~~~~很好看",
```

(续下页)

(接上页)

```
↪"我们到达的时候是下午3点，门票已经发完了。当时正焦虑的不知道怎么办才好，门卫大哥给我们俩补办了门票
↪",

↪"去过三次，个人认为这是长沙最值得去的地方，博物馆的重点就是辛追，遗憾的是，每次去我都会感到悲哀，虽
↪",

↪"上大学时候去的，当时学生证是半价25，后来凭有效证件就不要钱了。非常喜欢的一家博物馆，里面可看的东西
↪里面的讲解员大部分都是师大学历史类的，非常专业和有耐心。虽然不在长沙了，不过对那里还是很有感情的，
↪~~",

↪"这两年也有很多机会去博物馆。。。不过还是想说湖南省博物馆是非常有特色的。。。应该说整个展览分成两个
↪",

↪"网上订票去的，还是很顺利的就进去了，里面挺清净的，外围的环境也不错，还有鸽子可以喂。那天不是很闹，
↪",
]
encoder_tokenizer = BertTokenizer.from_pretrained("IDEA-CCNL/Randeng-GAVAE-1.2B-
↪Augmentation-Chinese")
decoder_tokenizer = T5Tokenizer.from_pretrained("IDEA-CCNL/Randeng-GAVAE-1.2B-
↪Augmentation-Chinese", eos_token = '<|endoftext|>', pad_token = '<pad>',extra_ids=0)
decoder_tokenizer.add_special_tokens({'bos_token':'<bos>'})
gavae_model = GAVAEModel.from_pretrained("IDEA-CCNL/Randeng-GAVAE-1.2B-Augmentation-
↪Chinese").to(device)
gavae_model.train_gan(encoder_tokenizer,decoder_tokenizer,input_texts)
# n:输出样本数量
texts = gavae_model.generate(n=5)
print(texts)
```

3.6.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的网站:

If you are using the resource for your work, please cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.7 Randeng-MegatronT5-770M

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.7.1 简介 Brief Introduction

善于处理 NLT 任务，中文版的 T5-large。

Good at solving NLT tasks, Chinese T5-large.

3.7.2 模型分类 Model Taxonomy

3.7.3 模型信息 Model Information

为了得到一个大规模的中文版的 T5，我们使用了 Megatron-LM 的方法和悟道语料库 (180G 版本) 用于预训练。具体地，我们在预训练阶段中使用了封神框架大概花费了 16 张 A100 约 14 天。

To get a large-scale Chinese T5, we use of Megatron-LM and WuDao Corpora (180 GB version) for pre-training. Specifically, we use the fengshen framework in the pre-training phase which cost about 14 days with 16 A100 GPUs.

3.7.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Randeng-MegatronT5-770M

加载模型 Loading Models

因为transformers库中是没有 Zhouwenwang-Unified-1.3B 相关的模型结构的，所以你可以在我们的Fengshenbang-LM中找到并且运行代码。

Since there is no structure of Randeng-MegatronT5-770M in transformers library, you can find the structure of Randeng-MegatronT5-770M and run the codes in Fengshenbang-LM.

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
```

```
from fengshen import T5ForConditionalGeneration
from fengshen import T5Config
from fengshen import T5Tokenizer
```

(续下页)

(接上页)

```
tokenizer = T5Tokenizer.from_pretrained('IDEA-CCNL/Randeng-MegatronT5-770M')
config = T5Config.from_pretrained('IDEA-CCNL/Randeng-MegatronT5-770M')
model = T5ForConditionalGeneration.from_pretrained('IDEA-CCNL/Randeng-MegatronT5-770M
↪')
```

之所以要进行上述操作是因为 T5 结构的 Randeng-MegatronT5-770M 模型是基于 Megatron 进行训练的，而 Megatron 的 T5 模型结构与 HuggingFace 的 T5 模型结构有略微的区别，不能直接使用 HuggingFace 的 T5 模型进行导入。因此需要从本仓库的 fengshen 框架导入，需要将 fengshen 放在你的工程文件夹。导入之后，即可按照下面的脚本从 HuggingFace 下载并加载对应的模型：

使用示例 Usage Examples

1、首先修改 finetune 示例脚本fengshen/scripts/finetune_classification.sh中的 model_type 和 pretrained_model_path 参数。其他如 batch_size、data_dir 等参数可根据自己的设备修改。

```
MODEL_TYPE=fengshen-megatron_t5
PRETRAINED_MODEL_PATH=IDEA-CCNL/Randeng-MegatronT5-770M
```

2、然后运行：

```
sh finetune_classification.sh
```

生成任务使用示例 Generation Examples

```
from fengshen import T5ForConditionalGeneration
from fengshen import T5Tokenizer

tokenizer = T5Tokenizer.from_pretrained('IDEA-CCNL/Randeng-MegatronT5-770M')
model = T5ForConditionalGeneration.from_pretrained('IDEA-CCNL/Randeng-MegatronT5-770M
↪')

output = model.generate(tokenizer.encode(tokenizer.encode('北京是中国的<extra_id_0>
↪')))
print(tokenizer.decode(output))
```

3.7.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.8 Randeng-Pegasus-238M-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.8.1 简介 Brief Introduction

善于处理摘要任务的，中文版的 PAGASUS-base。

Good at solving text summarization tasks, Chinese PAGASUS-base.

3.8.2 模型分类 Model Taxonomy

3.8.3 模型信息 Model Information

参考论文：PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization

为了解决中文的自动摘要任务，我们遵循 PEGASUS 的设计来训练中文的版本。我们使用了悟道语料库 (180G 版本) 作为预训练数据集。此外，考虑到中文 sentence piece 不稳定，我们在 Randeng-PEGASUS 中同时使用了结巴分词和 BERT 分词器。我们也提供 large 的版本：IDEA-CCNL/Randeng-Pegasus-523M-Chinese。以及，我们也提供了在中文摘要数据集上微调的版本：Randeng-Pegasus-238M-Summary-Chinese。

To solve Chinese abstractive summarization tasks, we follow the PEGASUS guidelines. We employ a version of WuDao Corpora (180 GB version) as a pre-training dataset. In addition, considering that the Chinese sentence chunk is unstable, we utilize jiebaand BERT tokenizer in our Randeng-PEGASUS. We also provide a large size version, available with IDEA-CCNL/Randeng-Pegasus-523M-Chinese. And, we also provide a version after fine-tuning on Chinese text summarization datasets: Randeng-Pegasus-238M-Summary-Chinese.

3.8.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Randeng-Pegasus-238M-Chinese

加载模型 Loading Models

```
from transformers import PegasusForConditionalGeneration
# Need to download tokenizers_pegasus.py and other Python script from Fengshenbang-LM␣
↪github repo in advance,
# or you can download tokenizers_pegasus.py and data_utils.py in https://huggingface.
↪co/IDEA-CCNL/Randeng_Pegasus_238M/tree/main
# Stronly recomend you git clone the Fengshenbang-LM repo:
# 1. git clone https://github.com/IDEA-CCNL/Fengshenbang-LM
# 2. cd Fengshenbang-LM/fengshen/examples/pegasus/
# and then you will see the tokenizers_pegasus.py and data_utils.py which are needed␣
↪by pegasus model
from tokenizers_pegasus import PegasusTokenizer

model = PegasusForConditionalGeneration.from_pretrained("IDEA-CCNL/Randeng-Pegasus-
↪238M-Chinese")
tokenizer = PegasusTokenizer.from_pretrained("IDEA-CCNL/Randeng-Pegasus-238M-Chinese")

text =
↪"据微信公众号“界面”报道，4日上午10点左右，中国发改委反垄断调查小组突击查访奔驰上海办事处，调取数
```

(续下页)

(接上页)

```
↪奔驰销售服务有限公司东区总经理在内的多名管理人员仍留在上海办公室内"
inputs = tokenizer(text, max_length=512, return_tensors="pt")

# Generate Summary
summary_ids = model.generate(inputs["input_ids"])
tokenizer.batch_decode(summary_ids, skip_special_tokens=True, clean_up_tokenization_
↪spaces=False)[0]
# model output: 截止昨日晚9点，包括北京梅赛德斯-
↪奔驰销售服务有限公司东区总经理在内的多名管理人员仍留在上海办公室内
```

3.8.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.9 Randeng-Pegasus-238M-Summary-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.9.1 简介 Brief Introduction

善于处理摘要任务，在数个中文摘要数据集上微调后的，中文版的 PAGASUS-base。

Good at solving text summarization tasks, after fine-tuning on multiple Chinese text summarization datasets, Chinese PAGASUS-base.

3.9.2 模型分类 Model Taxonomy

3.9.3 模型信息 Model Information

参考论文：PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization

基于Randeng-Pegasus-238M-Chinese，我们在收集的 7 个中文领域的文本摘要数据集（约 4M 个样本）上微调了它，得到了 summary 版本。这 7 个数据集为：education, new2016zh, nlpcc, shence, sohu, thucnews 和 weibo。

Based on Randeng-Pegasus-238M-Chinese, we fine-tuned a text summarization version (summary) on 7 Chinese text summarization datasets, with totaling around 4M samples. The datasets include: education, new2016zh, nlpcc, shence, sohu, thucnews and weibo.

下游效果 Performance

3.9.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Randeng-Pegasus-238M-Summary-Chinese

加载模型 Loading Models

```
from transformers import PegasusForConditionalGeneration,BertTokenizer
# Need to download tokenizers_pegasus.py and other Python script from Fengshenbang-LM␣
↪github repo in advance,
# or you can download tokenizers_pegasus.py and data_utils.py in https://huggingface.
↪co/IDEA-CCNL/Randeng_Pegasus_523M/tree/main
# Strongly recommend you git clone the Fengshenbang-LM repo:
```

(续下页)

(接上页)

```
# 1. git clone https://github.com/IDEA-CCNL/Fengshenbang-LM
# 2. cd Fengshenbang-LM/fengshen/examples/pegasus/
# and then you will see the tokenizers_pegasus.py and data_utils.py which are needed␣
↪by pegasus model

from tokenizers_pegasus import PegasusTokenizer

model = PegasusForConditionalGeneration.from_pretrained("IDEA-CCNL/Randeng-Pegasus-
↪238M-Summary-Chinese")
tokenizer = PegasusTokenizer.from_pretrained("IDEA-CCNL/Randeng-Pegasus-238M-Summary-
↪Chinese")

text =
↪"在北京冬奥会自由式滑雪女子坡面障碍技巧决赛中，中国选手谷爱凌夺得银牌。祝贺谷爱凌！今天上午，自由式
↪90分。在12位选手中排名第三。完成动作后，谷爱凌又扮了个鬼脸，甚是可爱。第二轮中，谷爱凌在道具区第三
↪98分。网友：摔倒了也没关系，继续加油！在第二跳失误摔倒的情况下，谷爱凌顶住压力，第三跳稳稳发挥，流
↪23分！此轮比赛，共12位选手参赛，谷爱凌第10位出场。网友：看比赛时我比谷爱凌紧张，加油！
↪"
inputs = tokenizer(text, max_length=1024, return_tensors="pt")

# Generate Summary
summary_ids = model.generate(inputs["input_ids"])
tokenizer.batch_decode(summary_ids, skip_special_tokens=True, clean_up_tokenization_
↪spaces=False)[0]

# model Output: 滑雪女子坡面障碍技巧决赛谷爱凌获银牌
```

3.9.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
```

(续下页)

(接上页)

```
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.10 Randeng-Pegasus-523M-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.10.1 简介 Brief Introduction

善于处理摘要任务的，中文版的 PAGASUS-large。

Good at solving text summarization tasks, Chinese PAGASUS-large.

3.10.2 模型分类 Model Taxonomy

3.10.3 模型信息 Model Information

参考论文：PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization

为了解决中文的自动摘要任务，我们遵循 PEGASUS 的设计来训练中文的版本。我们使用了悟道语料库 (180G 版本) 作为预训练数据集。此外，考虑到中文 sentence piece 不稳定，我们在 Randeng-PEGASUS 中同时使用了结巴分词和 BERT 分词器。我们也提供 base 的版本：IDEA-CCNL/Randeng-Pegasus-238M-Chinese。以及，我们也提供了在中文摘要数据集上微调的版本：Randeng-Pegasus-523M-Summary-Chinese。

To solve Chinese abstractive summarization tasks, we follow the PEGASUS guidelines. We employ a version of WuDao Corpora (180 GB version) as a pre-training dataset. In addition, considering that the Chinese sentence chunk is unstable, we utilize jieba and BERT tokenizer in our Randeng-PEGASUS. We also provide a base size version, available with IDEA-CCNL/Randeng-Pegasus-238M-Chinese. And, we also provide a version after fine-tuning on Chinese text summarization datasets: Randeng-Pegasus-523M-Summary-Chinese.

3.10.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Randeng-Pegasus-523M-Chinese

加载模型 Loading Models

```
from transformers import PegasusForConditionalGeneration
# Need to download tokenizers_pegasus.py and other Python script from Fengshenbang-LM␣
↪github repo in advance,
# or you can download tokenizers_pegasus.py and data_utils.py in https://huggingface.
↪co/IDEA-CCNL/Randeng_Pegasus_523M/tree/main
# Strongly recommend you git clone the Fengshenbang-LM repo:
# 1. git clone https://github.com/IDEA-CCNL/Fengshenbang-LM
# 2. cd Fengshenbang-LM/fengshen/examples/pegasus/
# and then you will see the tokenizers_pegasus.py and data_utils.py which are needed␣
↪by pegasus model
from tokenizers_pegasus import PegasusTokenizer

model = PegasusForConditionalGeneration.from_pretrained("IDEA-CCNL/Randeng-Pegasus-
↪523M-Chinese")
tokenizer = PegasusTokenizer.from_pretrained("IDEA-CCNL/Randeng-Pegasus-523M-Chinese")

text =
↪"据微信公众号“界面”报道，4日上午10点左右，中国发改委反垄断调查小组突击查访奔驰上海办事处，调取数
↪奔驰销售服务有限公司东区总经理在内的多名管理人员仍留在上海办公室内"
inputs = tokenizer(text, max_length=1024, return_tensors="pt")

# Generate Summary
summary_ids = model.generate(inputs["input_ids"])
tokenizer.batch_decode(summary_ids, skip_special_tokens=True, clean_up_tokenization_
↪spaces=False)[0]

# model Output: 截止昨日晚9点，包括北京梅赛德斯-
↪奔驰销售服务有限公司东区总经理在内的多名管理人员仍留在上海办公室内
```

3.10.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.11 Randeng-Pegasus-523M-Summary-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.11.1 简介 Brief Introduction

善于处理摘要任务，在数个中文摘要数据集上微调后的，中文版的 PAGASUS-large。

Good at solving text summarization tasks, after fine-tuning on multiple Chinese text summarization datasets, Chinese PAGASUS-large.

3.11.2 模型分类 Model Taxonomy

3.11.3 模型信息 Model Information

参考论文：PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization

基于Randeng-Pegasus-523M-Chinese，我们在收集的 7 个中文领域的文本摘要数据集（约 4M 个样本）上微调了它，得到了 summary 版本。这 7 个数据集为：education, new2016zh, nlpcc, shence, sohu, thucnews 和 weibo。

Based on Randeng-Pegasus-523M-Chinese, we fine-tuned a text summarization version (summary) on 7 Chinese text summarization datasets, with totaling around 4M samples. The datasets include: education, new2016zh, nlpcc, shence, sohu, thucnews and weibo.

微调细节 Details of Finetuning

finetune 的模型是燃灯模型，燃灯模型是 pegasus 结构，在预训练阶段主要是使用 wudao 数据进行的预训练，主要以中文语料为主。模型参数量总共为 5 亿，主要参数如下所示：

数据样例 Data Examples

用于 finetune 的 LCSTS 文本-标题对数据，格式如下：

```
.....
{'text': '.........', 'summary': '..........'}
{'text': '.........', 'summary': '..........'}
.....
```

Finetune 步骤

具体 fintune 代码在封神框架下，参考 fengshen/examples/summary/finetune_pegasus_summary.py 以及 randeng_pegasus_523M_summary.sh 两个脚本

1. 修改 randeng_pegasus_523M_summary.sh 脚本里的参数
2. 执行 sh randeng_pegasus_523M_summary.sh，即可开始 finetune

Finetune 参数

finetune 阶段使用了 deepspeed 来加速训练

其他训练参数请看 randeng_pegasus_523M_summary.sh

Finetune 后模型效果

LCSTS 摘要数据 finetune 后效果

3.11.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Randeng-Pegasus-523M-Summary-Chinese

加载模型 Loading Models

```
from transformers import PegasusForConditionalGeneration
# Need to download tokenizers_pegasus.py and other Python script from Fengshenbang-LM␣
↪github repo in advance,
# or you can download tokenizers_pegasus.py and data_utils.py in https://huggingface.
↪co/IDEA-CCNL/Randeng_Pegasus_523M/tree/main
# Strongly recommend you git clone the Fengshenbang-LM repo:
# 1. git clone https://github.com/IDEA-CCNL/Fengshenbang-LM
# 2. cd Fengshenbang-LM/fengshen/examples/pegasus/
# and then you will see the tokenizers_pegasus.py and data_utils.py which are needed␣
↪by pegasus model
from tokenizers_pegasus import PegasusTokenizer

model = PegasusForConditionalGeneration.from_pretrained("IDEA-CCNL/Randeng-Pegasus-
↪523M-Summary-Chinese")
tokenizer = PegasusTokenizer.from_pretrained("IDEA-CCNL/Randeng-Pegasus-523M-Summary-
↪Chinese")

text =
↪"据微信公众号“界面”报道，4日上午10点左右，中国发改委反垄断调查小组突击查访奔驰上海办事处，调取数
↪奔驰销售服务有限公司东区总经理在内的多名管理人员仍留在上海办公室内"
inputs = tokenizer(text, max_length=1024, return_tensors="pt")

# Generate Summary
summary_ids = model.generate(inputs["input_ids"])
```

(续下页)

(接上页)

```
tokenizer.batch_decode(summary_ids, skip_special_tokens=True, clean_up_tokenization_
↪spaces=False)[0]

# model Output: 反垄断调查小组突击查访奔驰上海办事处，对多名奔驰高管进行约谈
```

3.11.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.12 Randeng-PPVAE-1.2B-Augmentation-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.12.1 简介 Brief Introduction

PPVAE(Pre-train and Plug-in Variational Auto-Encoder) 可以通过少量类别文本的训练生成大量该类别的增强样本。PPVAE 是一个由两个 VAE 组成的层级框架：预训练 VAE 的编码器得到文本全局隐空间，解码器将隐向量解码为文本；PluginVAE 为一个轻量级 VAE，学习从全局隐空间到条件隐空间的相互映射，该映射只需要少量条件文本即可训练完成。

PPVAE (Pre-train and Plug-in Variational Auto-Encoder) can generate a large number of category-specific samples from the training of a small number of category texts. PPVAE is a hierarchical framework consisting of two VAEs: the encoder of the pre-trained VAE gets the text global hidden space and the decoder decodes the hidden vector into text; PluginVAE is a lightweight VAE that learns the transformation from the global hidden space to the conditional hidden space, which requires only a small amount of conditional text to be trained.

PPVAE 参考论文Pre-train and Plug-in: Flexible Conditional Text Generation with Variational Auto-Encoders.

PPVAE reference paper Pre-training and Plug-in: Flexible Conditional Text Generation with Variable Autoencoders.

3.12.2 模型分类 Model Taxonomy

3.12.3 模型信息 Model Information

Pretrained VAE:

训练语料：悟道语料库（280G 版本）

Training Corpus: Wudao Corpus (with 280G samples)

参考模型：Randeng-DAVAE-1.2B-General-Chinese

Reference model:Randeng-DAVAE-1.2B-General-Chinese

PluginVAE:

编码器：三层 MLP，将隐向量从全局隐空间映射到类别隐空间；

解码器：三层 MLP，将隐向量从类别隐空间映射到全局隐空间。

训练语料：少量类别文本。

Encoder: three-layer MLP that maps the hidden vector from the global hidden space to the category hidden space.

Decoder: three-layer MLP, mapping hidden vectors from the category hidden space to the global hidden space.

Training corpus: a small amount of categorical text.

3.12.4 使用 Usage

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
cd Fengshenbang-LM
pip install --editable .
```

```
import torch
from transformers import BertTokenizer,T5Tokenizer
from fengshen.models.PPVAE.pluginVAE import PPVAEModel
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
input_texts = [
    "非常好的一个博物馆，是我所有去过的博物馆里感觉最正规的一家.",
    "这是我来长沙最最期待的一定要去的地方，总算今天特地去瞻仰千古遗容了，真好。",
    "地方很大 很气派~~可以逛很久~~~去的时候是免费的~不过要安检~~~里面的马王堆~
↪幸追夫人~还是很不错的",

↪"绝对不虚此行！相当震撼的展览！原来古人也化妆，还有假发。记忆最深的是那个藕汤。可惜真颜已不得见。
↪",
    "去过三次，个人认为这是长沙最值得去的地方.",

↪"非常喜欢的一家博物馆，里面可看的东西很多，当然最吸引我的就是那个辛追夫人和“素纱单衣”，果然不是盖
↪赞~~~",

↪"这两年也有很多机会去博物馆。。。不过还是想说湖南省博物馆是非常有特色的。。。真是上了一节很生动的历
↪",

↪"网上订票去的，还是很顺利的就进去了，里面挺清净的，外围的环境也不错，还有鸽子可以喂。
↪",
]
encoder_tokenizer = BertTokenizer.from_pretrained("IDEA-CCNL/Randeng-PPVAE-1.2B-
↪General-Chinese")
decoder_tokenizer = T5Tokenizer.from_pretrained("IDEA-CCNL/Randeng-PPVAE-1.2B-General-
↪Chinese", eos_token = '<|endoftext|>', pad_token = '<pad>',extra_ids=0)
decoder_tokenizer.add_special_tokens({'bos_token':'<bos>'})
ppvae_model = PPVAEModel.from_pretrained("IDEA-CCNL/Randeng-PPVAE-1.2B-Augmentation-
↪Chinese").to(device)
ppvae_model.train_plugin(encoder_tokenizer,decoder_tokenizer,input_texts,negative_
↪samples=None)
# n:输出样本数量
texts = ppvae_model.generate(n=5)
print(texts)
# 生成结果样例:
# ['同学很推荐那里,自然会有好的风景.那里物价很便宜,真的不错。',
# '同学说一会去盛国,可能是我去的比较多!故居真的很漂亮,夜景也特别好看。'
```

(续下页)

(接上页)

```
# '我的第一次旅行没有白来,最后领略了有些风吹草低见牛羊的味道,谢谢本次疗养。',
# '同学一打听:这里距离世纪公园,还有最近的香山营不过200米,海拔也才四千米。',
# '我发现那边很文艺!!有机会去过的,真是土耳其当地口音~还是很干净!。', ]
```

3.12.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的网站:

If you are using the resource for your work, please cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.13 Randeng-Transformer-1.1B-Denoise

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.13.1 简介 Brief Introduction

以去噪任务为微调目标的中文 Transformer-XL。

Chinese Transformer-XL with a denoising task as the fine-tuning objective.

3.13.2 模型分类 Model Taxonomy

3.13.3 模型信息 Model Information

我们先使用 Transformer-XL 的模型结构在悟道语料库 (180G 版本) 上进行预训练，然后在我们自主构建的去噪数据集上进行微调。其中，去噪任务是从包括 **随机插入/交换/删除/替换/句子重排**的具有噪声的输入中重建一个流畅和干净的文本。

We first pre-trained Transformer-XL on the Wudo corpus (180G version), and then fine-tuned it on a denoised dataset (developed by us). The denoise task is to reconstruct a fluent and clean text from a noisy input which includes **random insertion/swap/deletion/replacement/sentence reordering**.

3.13.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Randeng-Transformer-1.1B-Denoise

加载模型 Loading Models

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
```

```
from fengshen.models.transfo_xl_denoise.tokenization_transfo_xl_denoise import␣
↪TransfoXLDenoiseTokenizer
from fengshen.models.transfo_xl_denoise.modeling_transfo_xl_denoise import␣
↪TransfoXLDenoiseModel

tokenizer = TransfoXLDenoiseTokenizer.from_pretrained('IDEA-CCNL/Randeng-Transformer-
↪1.1B-Denoise')
model = TransfoXLDenoiseModel.from_pretrained('IDEA-CCNL/Randeng-Transformer-1.1B-
↪Denoise')
```

使用示例 Usage Examples

```
from fengshen.models.transfo_xl_denoise.generate import denoise_generate
input_text = "凡是有成就的人，都很严肃地对待生命自己的"
res = denoise_generate(model, tokenizer,  input_text)
print(res)
# "有成就的人都很严肃地对待自己的生命。"
```

3.13.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
```

(续下页)

(接上页)

```
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.14 Randeng-TransformerXL-5B-Deduction-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs
- Demo: Reasoning Tree

3.14.1 简介 Brief Introduction

基于 Transformer-XL 的中文因果推理生成模型。

Chinese deductive reasoning model based on Transformer-XL.

3.14.2 模型分类 Model Taxonomy

3.14.3 模型信息 Model Information

数据准备 Corpus Preparation

- 悟道语料库（280G 版本）
- 因果语料库（2.3M 个样本）：基于悟道语料库（280G 版本），通过关联词匹配、人工标注 + GTSFactory筛选、数据清洗等步骤获取的具有因果关系的句子对
- Wudao Corpus (with 280G samples)

- Wudao Causal Corpus (with 2.3 million samples): Based on the Wudao corpus (280G version), sentence pairs with causality were obtained through logic indicator matching, manual annotation + GTSFactory, and data cleaning.

训练流程 Model Training

1. 在悟道语料库（280G 版本）上进行预训练
2. 在 1.5M 因果语料上进行因果生成任务的训练
3. 基于其余 0.8M 因果语料，协同Randeng-TransformerXL-5B-Abduction-Chinese和Erlangshen-Roberta-330M-Causal-Chinese进行 Self-consistent 闭环迭代训练
   - 两个生成模型基于核采样和贪心的方式进行因果推理和反绎推理，产生大量伪样本；
   - Erlangshen-Roberta-330M-Causal-Chinese 模型对伪样本句子对的因果关系进行打分，筛选供自身以及生成模型训练的样本

First, the Transformer-XL model was pre-trained on the Wudao Corpus (with 280G samples) and annotated similar-sentence pair dataset (same as Randeng-TransformerXL-1.1B-Paraphrasing-Chinese). Then, the model was trained on our causal corpus (about 1.5 million samples) for the deductive reasoning task. At last, based on the remaining 0.8 million samples of the causal corpus, we conducted self-consistent learning on this model, cooperating with Randeng-TransformerXL-5B-Abduction-Chinese and Erlangshen-Roberta-330M-Causal-Chinese. Specifically, two generative models performed deductive reasoning and abductive reasoning based on each sample respectively, generating a large number of pseudo-samples; Erlangshen-Roberta-330M-Causal-Chinese scored the causality of the pseudo-samples and selected the training data for itself and the generative models in the next iteration.

3.14.4 加载模型 Loading Models

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
cd Fengshenbang-LM
```

```
from fengshen.models.transfo_xl_reasoning import TransfoXLModel
from transformers import T5Tokenizer as TransfoXLTokenizer

model = TransfoXLModel.from_pretrained('IDEA-CCNL/Randeng-TransformerXL-5B-Deduction-
↪Chinese')
tokenizer = TransfoXLTokenizer.from_pretrained(
    "IDEA-CCNL/Randeng-TransformerXL-5B-Deduction-Chinese",
    eos_token='<|endoftext|>',
    pad_token='<|endoftext|>',
    extra_ids=0
)
tokenizer.add_special_tokens({'bos_token': '<bos>'})
```

3.14.5 使用示例 Usage Example

```
from fengshen.models.transfo_xl_reasoning import deduction_generate

input_text = "机器人统治世界"
input_texts = ["机器人统治世界", "玉米价格持续上涨"]
print(deduction_generate(model, tokenizer, input_text, device=0))
print(deduction_generate(model, tokenizer, input_texts, device=0))
```

3.14.6 一些例子 Some Cases

（以下内容为模型生成，不代表本单位观点）

1. 鲸鱼是哺乳动物，哺乳动物是恒温动物
    - 鲸鱼拥有极强的恒温能力
    - 鲸鱼需要消耗大量的能量
    - 鲸鱼能在海洋中生存下来，完全依靠其体温恒定
2. 鲸鱼是哺乳动物，哺乳动物不是恒温动物
    - 鲸鱼不是恒温动物，鲸鱼是变温动物
    - 鲸鱼会随温度变化而变化
    - 鲸鱼有冬眠现象
3. 机器人统治世界
    - 机器人取代人类做大部分工作
    - 机器人世界的法律必须能够适用于机器人统治的现实世界
    - 机器人必须能够相互沟通，并能够处理人类无法处理的复杂情况
    - 未来将不再需要大量工人，机器人将接管工厂
4. 玉米价格持续上涨
    - 玉米淀粉价格也呈现上涨趋势
    - 玉米种植效益不断攀升
    - 在玉米深加工行业引起了一阵骚动
5. 实体经济融资难、融资贵
    - 急需发展互联网金融等金融业态，为实体经济提供融资服务
    - 融资需求向金融资产转移，增加了金融资产供给
    - 必须大力发展资本市场，使资本市场成为经济转型的助推器

6. 影响华北地区的冷空气势力偏弱
    - 冷空气的影响时间将偏短
    - 冷空气影响结束后，华北地区气温会继续缓慢回升
    - 华北地区气温较常年同期偏高

3.14.7 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.15 Randeng-TransformerXL-5B-Abduction-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs
- Demo: Reasoning Tree

3.15.1 简介 Brief Introduction

基于 Transformer-XL 的中文反绎（溯因）推理生成模型。

Chinese abductive reasoning model based on Transformer-XL.

3.15.2 模型分类 Model Taxonomy

3.15.3 模型信息 Model Information

数据准备 Corpus Preparation

- 悟道语料库（280G 版本）
- 因果语料库（2.3M 个样本）：基于悟道语料库（280G 版本），通过关联词匹配、人工标注 + GTSFactory筛选、数据清洗等步骤获取的具有因果关系的句子对
- Wudao Corpus (with 280G samples)
- Wudao Causal Corpus (with 2.3 million samples): Based on the Wudao corpus (280G version), sentence pairs with causality were obtained through logic indicator matching, manual annotation + GTSFactory, and data cleaning.

训练流程 Model Training

1. 在悟道语料库（280G 版本）上进行预训练
2. 在 1.5M 因果语料上进行反绎生成任务的训练
3. 基于其余 0.8M 因果语料，协同Randeng-TransformerXL-5B-Deduction-Chinese和Erlangshen-Roberta-330M-Causal-Chinese进行 Self-consistent 闭环迭代训练
    - 两个生成模型基于核采样和贪心的方式进行因果推理和反绎推理，产生大量伪样本；
    - Erlangshen-Roberta-330M-Causal-Chinese 模型对伪样本句子对的因果关系进行打分，筛选供自身以及生成模型训练的样本

First, the Transformer-XL model was pre-trained on the Wudao Corpus (with 280G samples) and annotated similar-sentence pair dataset (same as Randeng-TransformerXL-1.1B-Paraphrasing-Chinese). Then, the model was trained on our causal corpus (about 1.5 million samples) for the abductive reasoning task. At last, based on the remaining 0.8 million samples of the causal corpus, we conducted self-consistent learning on this model, cooperating with Randeng-TransformerXL-5B-Deduction-Chinese and Erlangshen-Roberta-330M-Causal-Chinese. Specifically, two generative models performed deductive reasoning and abductive reasoning based on each sample respectively, generating a large number of pseudo-samples; Erlangshen-Roberta-330M-Causal-Chinese scored the causality of the pseudo-samples and selected the training data for itself and the generative models in the next iteration.

3.15.4 加载模型 Loading Models

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
cd Fengshenbang-LM
```

```
from fengshen.models.transfo_xl_reasoning import TransfoXLModel
from transformers import T5Tokenizer as TransfoXLTokenizer

model = TransfoXLModel.from_pretrained('IDEA-CCNL/Randeng-TransformerXL-5B-Abduction-
↪Chinese')
tokenizer = TransfoXLTokenizer.from_pretrained(
    "IDEA-CCNL/Randeng-TransformerXL-5B-Abduction-Chinese",
    eos_token='<|endoftext|>',
    pad_token='<|endoftext|>',
    extra_ids=0
)
tokenizer.add_special_tokens({'bos_token': '<bos>'})
```

3.15.5 使用示例 Usage Example

```
from fengshen.models.transfo_xl_reasoning import abduction_generate

input_text = "玉米价格持续上涨"
input_texts = ["玉米价格持续上涨", "玉米价格持续上涨"]
print(abduction_generate(model, tokenizer, input_text, device=0))
print(abduction_generate(model, tokenizer, input_texts, device=0))
```

3.15.6 一些例子 Some Cases

（以下内容为模型生成，不代表本单位观点）

1. 玉米价格持续上涨
    - 玉米库存较低，需求增加
    - 东北地区受降雨天气影响，玉米生长受到影响
    - 今年玉米种植面积大幅度下降
2. 玉米价格下跌
    - 玉米的库存量大，需求量低
    - 今年玉米产量创新高，而需求不足
    - 目前玉米市场处于供大于求的状态，再加上近期华北地区遭遇了强降雨天气，玉米质量下降

3. 农作物大量死亡
    - 旱灾持续时间长，又无雨，土壤干裂，作物得不到水分
    - 霜冻来临，气温骤降，植物受冻
    - 许多农民为了使农作物能够长得更好，使用更多的农药，并且没有合理的休耕措施
4. 鲸鱼需要消耗大量的能量
    - 鲸鱼的体型庞大，新陈代谢速度又快
    - 鲸鱼的身体结构特殊，需要消耗大量的能量来维持身体结构的稳定
5. 实体经济融资难、融资贵
    - 融资渠道单一，实体经济难以获得充足的资金
    - 实体经济融资主要依赖抵押、担保、信贷等间接融资方式，存在抵押物不足、担保机制不完善等问题
    - 实体经济往往需要大量的资金，而银行受制于风险控制、资本充足率等要求，很难大量发放贷款
6. 火山爆发导致植物死亡
    - 火山灰会阻碍植物吸收阳光
    - 火山灰的飘散，导致植物无法吸收到足够的氧气
    - 火山喷发时，岩浆温度极高，植物无法承受

3.15.7 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.16 Randeng-T5-77M

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.16.1 简介 Brief Introduction

善于处理 NLT 任务，中文版的 mT5-small。

Good at handling NLT tasks, Chinese mT5-small.

3.16.2 模型分类 Model Taxonomy

3.16.3 模型信息 Model Information

我们基于 mT5-small，训练了它的中文版。为了加速训练，我们仅使用 T5 分词器 (sentence piece) 中的中英文对应的词表，并且使用了语料库自适应预训练 (Corpus-Adaptive Pre-Training, CAPT) 技术在悟道语料库 (180G 版本) 继续预训练。预训练目标为破坏 span。具体地，我们在预训练阶段中使用了封神框架大概花费了 8 张 A100 约 24 小时。

Based on mT5-small, we implement its Chinese version. In order to accelerate training, we only retrain the vocabulary and embedding corresponding to Chinese and English in T5tokenizer (sentence piece), and Corpus-Adaptive Pre-Training (CAPT) on the WuDao Corpora (180 GB version). The pretraining objective is span corruption. Specifically, we use the fengshen framework in the pre-training phase which cost about 24 hours with 8 A100 GPUs.

3.16.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Randeng-T5-77M

加载模型 Loading Models

```
from transformers import T5ForConditionalGeneration, AutoTokenizer
import torch

tokenizer=AutoTokenizer.from_pretrained('IDEA-CCNL/Randeng-T5-77M', use_fast=false)
model=T5ForConditionalGeneration.from_pretrained('IDEA-CCNL/Randeng-T5-77M')
```

数据处理 Data Processing

通用的数据举例：

```
text: '运动,
↪走势在什么时候结束是不可能有答案的。为了找到走势什么时候结束原来的运动方向而改变方向,
↪必须引进新的概念:中枢。'
```

span corruption 后的数据举例

```
input: '运动,走势在什么时候结束是不可能有答案的。为了 <extra_id_0>
↪走势什么时候结束原来 <extra_id_1>必须引进新的概念:中枢。'

label: '<extra_id_0>找到 <extra_id_1>的运动方向而改变方向,\</s>'
```

对应的代码见：地址。

模型训练 Training

模型利用封神框架在 2 张 A100 训练 17 小时，最后 loss 收敛到 2.3 左右，训练脚本见：地址。

3.16.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
```

(续下页)

(接上页)

```
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

3.17 Randeng-T5-784M

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

3.17.1 简介 Brief Introduction

善于处理 NLT 任务，中文版的 mT5-large。

Good at handling NLT tasks, Chinese mT5-large.

3.17.2 模型分类 Model Taxonomy

3.17.3 模型信息 Model Information

我们基于 mT5-large，训练了它的中文版。为了加速训练，我们仅使用 T5 分词器 (sentence piece) 中的中英文对应的词表，并且使用了语料库自适应预训练 (Corpus-Adaptive Pre-Training, CAPT) 技术在悟道语料库 (180G 版本) 继续预训练。预训练目标为破坏 span。具体地，我们在预训练阶段中使用了封神框架大概花费了 16 张 A100 约 96 小时。

Based on mT5-large, we implement its Chinese version. In order to accelerate training, we only retrain the vocabulary and embedding corresponding to Chinese and English in T5tokenizer (sentence piece), and Corpus-Adaptive Pre-Training (CAPT) on the WuDao Corpora (180 GB version). The pretraining objective is span corruption. Specifically, we use the fengshen framework in the pre-training phase which cost about 96 hours with 16 A100 GPUs.

3.17.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Randeng-T5-784M

加载模型 Loading Models

```
from transformers import T5ForConditionalGeneration, AutoTokenizer
import torch
tokenizer=AutoTokenizer.from_pretrained('IDEA-CCNL/Randeng-T5-784M', use_fast=false)
model=T5ForConditionalGeneration.from_pretrained('IDEA-CCNL/Randeng-T5-784M')
```

数据处理 Data Processing

(同 Randeng-T5-77M)

通用的数据举例：

```
text: '运动,
↪走势在什么时候结束是不可能有答案的。为了找到走势什么时候结束原来的运动方向而改变方向,
↪必须引进新的概念:中枢。'
```

span corruption 后的数据举例

```
input: '运动,走势在什么时候结束是不可能有答案的。为了 <extra_id_0>
↪走势什么时候结束原来 <extra_id_1>必须引进新的概念:中枢。'

label: '<extra_id_0>找到 <extra_id_1>的运动方向而改变方向,\</s>'
```

对应的代码见：Fengshenbang-LM/fengshen/data/t5_dataloader/t5_datasets.py

对应的代码见：地址。

模型训练 Training

可以参考 Randeng-T5-77M 的训练脚本：地址。

3.17.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

CHAPTER 4

太乙系列

4.1 Taiyi-CLIP-Roberta-102M-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

4.1.1 简介 Brief Introduction

首个开源的中文 CLIP 模型，1.23 亿图文对上进行预训练的文本端 RoBERTa-base。

The first open source Chinese CLIP, pre-training on 123M image-text pairs, the text encoder: RoBERTa-base.

4.1.2 模型分类 Model Taxonomy

4.1.3 模型信息 Model Information

我们遵循 CLIP 的实验设置，以获得强大的视觉-语言表征。在训练中文版的 CLIP 时，我们使用chinese-roberta-wwm作为语言的编码器，并将CLIP中的 ViT-B-32 应用于视觉的编码器。为了快速且稳定地进行预训练，我们冻结了视觉编码器并且只微调语言编码器。此外，我们将Noah-Wukong数据集 (100M) 和Zero数据集 (23M) 用作预训练的数据集。据我们所知，我们的 Taiyi-CLIP 是目前 Huggingface 社区中首个的开源中文 CLIP。

We follow the experimental setup of CLIP to obtain powerful visual-language intelligence. To obtain the CLIP for Chinese, we employ chinese-roberta-wwm for the language encoder, and apply the ViT-B-32 in CLIP for the vision encoder. We freeze the vision encoder and tune the language encoder to speed up and stabilize the pre-training process.

Moreover, we apply Noah-Wukong dataset (100M) and Zero dataset (23M) as the pre-training datasets. To the best of our knowledge, our TaiyiCLIP is currently the only open-sourced Chinese CLIP in the huggingface community.

下游效果 Performance

Zero-Shot Classification

Zero-Shot Text-to-Image Retrieval

4.1.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Taiyi-CLIP-Roberta-102M-Chinese

加载模型 Loading Models

```
from PIL import Image
import requests
import clip
import torch
from transformers import BertForSequenceClassification, BertConfig, BertTokenizer
from transformers import CLIPProcessor, CLIPModel
import numpy as np

query_texts = ["一只猫", "一只狗",'两只猫', '两只老虎','一只老虎']  #␣
↪这里是输入文本的，可以随意替换。
# 加载Taiyi 中文 text encoder
text_tokenizer = BertTokenizer.from_pretrained("IDEA-CCNL/Taiyi-CLIP-Roberta-102M-
↪Chinese")
text_encoder = BertForSequenceClassification.from_pretrained("IDEA-CCNL/Taiyi-CLIP-
↪Roberta-102M-Chinese").eval()
text = text_tokenizer(query_texts, return_tensors='pt', padding=True)['input_ids']

url = "http://images.cocodataset.org/val2017/000000039769.jpg"  #␣
↪这里可以换成任意图片的url
# 加载CLIP的image encoder
clip_model = CLIPModel.from_pretrained("openai/clip-vit-base-patch32")
processor = CLIPProcessor.from_pretrained("openai/clip-vit-base-patch32")
image = processor(images=Image.open(requests.get(url, stream=True).raw), return_
↪tensors="pt")
if image_data.mode != 'RGB':
    image = image.convert('RGB')
```

(续下页)

(接上页)

```
with torch.no_grad():
    image_features = clip_model.get_image_features(**image)
    text_features = text_encoder(text).logits
    # 归一化
    image_features = image_features / image_features.norm(dim=1, keepdim=True)
    text_features = text_features / text_features.norm(dim=1, keepdim=True)
    # 计算余弦相似度 logit_scale是尺度系数
    logit_scale = clip_model.logit_scale.exp()
    logits_per_image = logit_scale * image_features @ text_features.t()
    logits_per_text = logits_per_image.t()
    probs = logits_per_image.softmax(dim=-1).cpu().numpy()
    print(np.around(probs, 3))
```

在下游任务微调 Finetuning

我们提供了 CLIP 在 Flickr30k-CNA 这个数据集上的 finetune 代码示例，另外我们也提供了召回率计算的代码，都集成在 LightningModule 里了。

具体见：地址

配置好相关环境后，执行 `sh finetune_flickr.sh` 即可。

4.1.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

4.2 Taiyi-CLIP-Roberta-large-326M-Chinese

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

4.2.1 简介 Brief Introduction

首个开源的中文 CLIP 模型，1.23 亿图文对上进行预训练的文本端 RoBERTa-large。

The first open source Chinese CLIP, pre-training on 123M image-text pairs, the text encoder: RoBERTa-large.

4.2.2 模型分类 Model Taxonomy

4.2.3 模型信息 Model Information

我们遵循 CLIP 的实验设置，以获得强大的视觉-语言表征。在训练中文版的 CLIP 时，我们使用chinese-roberta-wwm-large作为语言的编码器，并将CLIP中的 **ViT-L-14** 应用于视觉的编码器。为了快速且稳定地进行预训练，我们冻结了视觉编码器并且只微调语言编码器。此外，我们将Noah-Wukong数据集 (100M) 和Zero数据集 (23M) 用作预训练的数据集。我们先在悟空数据集上预训练了 10 轮，然后接着在悟空数据集和 zero 数据集上预训练 12 轮。据我们所知，我们的 Taiyi-CLIP 是目前 Huggingface 社区中首个的开源中文 CLIP。

We follow the experimental setup of CLIP to obtain powerful visual-language intelligence. To obtain the CLIP for Chinese, we employ chinese-roberta-wwm-large for the language encoder, and apply the **ViT-L-14** in CLIP for the vision encoder. We freeze the vision encoder and tune the language encoder to speed up and stabilize the pre-training process. Moreover, we apply Noah-Wukong dataset (100M) and Zero dataset (23M) as the pre-training datasets. The model was first trained 10 epochs on wukong and then train another 12 epochs on wukong and zero. To the best of our knowledge, our TaiyiCLIP is currently the only open-sourced Chinese CLIP in the huggingface community.

下游效果 Performance

Zero-Shot Classification

Zero-Shot Text-to-Image Retrieval

4.2.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Taiyi-CLIP-Roberta-large-326M-Chinese

加载模型 Loading Models

```
from PIL import Image
import requests
import clip
import torch
from transformers import BertForSequenceClassification, BertConfig, BertTokenizer
from transformers import CLIPProcessor, CLIPModel
import numpy as np

query_texts = ["一只猫", "一只狗",'两只猫', '两只老虎','一只老虎']  #␣
↪这里是输入文本的，可以随意替换。
# 加载Taiyi 中文 text encoder
text_tokenizer = BertTokenizer.from_pretrained("IDEA-CCNL/Taiyi-CLIP-Roberta-large-
↪326M-Chinese")
text_encoder = BertForSequenceClassification.from_pretrained("IDEA-CCNL/Taiyi-CLIP-
↪Roberta-large-326M-Chinese").eval()
text = text_tokenizer(query_texts, return_tensors='pt', padding=True)['input_ids']

url = "http://images.cocodataset.org/val2017/000000039769.jpg"  #␣
↪这里可以换成任意图片的url
# 加载CLIP的image encoder
clip_model = CLIPModel.from_pretrained("openai/clip-vit-large-patch14")
processor = CLIPProcessor.from_pretrained("openai/clip-vit-large-patch14")
image = processor(images=Image.open(requests.get(url, stream=True).raw), return_
↪tensors="pt")
if image_data.mode != 'RGB':
    image = image.convert('RGB')

with torch.no_grad():
    image_features = clip_model.get_image_features(**image)
```

(续下页)

(接上页)

```
    text_features = text_encoder(text).logits
    # 归一化
    image_features = image_features / image_features.norm(dim=1, keepdim=True)
    text_features = text_features / text_features.norm(dim=1, keepdim=True)
    # 计算余弦相似度 logit_scale是尺度系数
    logit_scale = clip_model.logit_scale.exp()
    logits_per_image = logit_scale * image_features @ text_features.t()
    logits_per_text = logits_per_image.t()
    probs = logits_per_image.softmax(dim=-1).cpu().numpy()
    print(np.around(probs, 3))
```

在下游任务微调 Finetuning

我们提供了 CLIP 在 Flickr30k-CNA 这个数据集上的 finetune 代码示例，另外我们也提供了召回率计算的代码，都集成在 LightningModule 里了。(案例是 base 版的，直接替换模型就可以用 large 版来 finetune)

具体见：地址

配置好相关环境后，执行 `sh finetune_flickr.sh` 即可。

4.2.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

4.3 Taiyi-Roberta-124M-D

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

4.3.1 简介 Brief Introduction

使用了 4M 图文对进行特殊预训练的，英文版的 MAP（名称暂定）的文本端 RoBERTa-base。

Special pre-training on 1M image-text pairs, the textual encoder for MAP (temporary) in English, RoBERTa-base.

4.3.2 模型分类 Model Taxonomy

4.3.3 模型信息 Model Information

基于 Roberta-base，我们使用特殊的训练任务引入一些多模态信息。”D” 表示这是一种新的预训练方法。对于特殊的多模态表征，在论文中我们设计了集中不同的训练目标。预训练数据集为 MSCOCO,VG 和 SBU。我们的代码和预训练任务的细节将在论文接受后公开。

Based on pre-trained Roberta-base, we apply some multimodal information with special pre-training tasks. ”D” implies a special training method. For special multimodal representations, we design several special training objectives in our paper. The pre-training datasets are MSCOCO, VG and SBU. Our code and details of pre-training tasks will be made publicly available upon paper acceptance.

下游效果 Performance

GLUE

The local test settings are: Sequence length: 128, Batch size: 32, Learning rate: 3e-5

4.3.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Taiyi-Roberta-124M-D-v2

加载模型 Loading Models

```
from transformers import RobertaTokenizer, RobertaModel

tokenizer = RobertaTokenizer.from_pretrained("IDEA-CCNL/Taiyi-Roberta-124M-D-v2")
model = RobertaModel.from_pretrained("IDEA-CCNL/Taiyi-Roberta-124M-D-v2")
```

4.3.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

4.4 Taiyi-Roberta-124M-D

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

4.4.1 简介 Brief Introduction

COCO 和 VG 上特殊预训练的，英文版的 MAP（名称暂定）的文本端 RoBERTa-base。

Special pre-training on COCO and VG, the textual encoder for MAP (temporary) in English, RoBERTa-base.

4.4.2 模型分类 Model Taxonomy

4.4.3 模型信息 Model Information

基于 Roberta-base，我们使用特殊的训练任务引入一些多模态信息。”D” 表示这是一种新的预训练方法。对于特殊的多模态表征，在论文中我们设计了集中不同的训练目标。预训练数据集为 MSCOCO 和 VG。我们的代码和预训练任务的细节将在论文接受后公开。

Based on pre-trained Roberta-base, we apply some multimodal information with special pre-training tasks. ”D” implies a special training method. For special multimodal representations, we design several special training objectives in our paper. The pre-training datasets are MSCOCO and VG. Our code and details of pre-training tasks will be made publicly available upon paper acceptance.

下游效果 Performance

GLUE

The local test settings are: Sequence length: 128, Batch size: 32, Learning rate: 3e-5

An additional dataset WNLI is tested.

4.4.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Taiyi-Roberta-124M-D

加载模型 Loading Models

```
from transformers import RobertaTokenizer, RobertaModel

tokenizer = RobertaTokenizer.from_pretrained("IDEA-CCNL/Taiyi-Roberta-124M-D")
model = RobertaModel.from_pretrained("IDEA-CCNL/Taiyi-Roberta-124M-D")
```

4.4.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

4.5 Taiyi-vit-87M-D

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

4.5.1 简介 Brief Introduction

COCO 和 VG 上特殊预训练的，英文版的 MAP（名称暂定）的视觉端 ViT-base。

Special pre-training on COCO and VG, the visual encoder for MAP (temporary) in English, ViT-base.

4.5.2 模型分类 Model Taxonomy

4.5.3 模型信息 Model Information

基于 clip-vit-base (patch 16, resolution 224x224)，我们使用特殊的训练任务引入一些多模态信息。”D” 表示这是一种新的预训练方法。对于特殊的多模态表征，在论文中我们设计了集中不同的训练目标。预训练数据集为 MSCOCO 和 VG。我们的代码和预训练任务的细节将在论文接受后公开。

Based on pre-trained clip-vit-base (patch 16, resolution 224x224), we apply some multimodal information with special pre-training tasks. ”D” implies a special training method. For special multimodal representations, we design several special training objectives in our paper. The pre-training datasets are MSCOCO and VG. Our code and details of pre-training tasks will be made publicly available upon paper acceptance.

下游任务 Performance

The local test settings are:

learning rate=2e-5, batch size=128, num train epochs=5, weight decay=0.01

4.5.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Taiyi-vit-87M-D

加载模型 Loading Models

```
from transformers import ViTFeatureExtractor, ViTForImageClassification
from PIL import Image
import requests

url = 'http://images.cocodataset.org/val2017/000000039769.jpg'
image = Image.open(requests.get(url, stream=True).raw)

feature_extractor = ViTFeatureExtractor.from_pretrained('IDEA-CCNL/Taiyi-vit-87M-D')
model = ViTForImageClassification.from_pretrained('IDEA-CCNL/Taiyi-vit-87M-D')

inputs = feature_extractor(images=image, return_tensors="pt")
outputs = model(**inputs)
logits = outputs.logits
# model predicts one of the 1000 ImageNet classes
predicted_class_idx = logits.argmax(-1).item()
print("Predicted class:", model.config.id2label[predicted_class_idx])
# Predicted class: Egyptian cat
```

4.5.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

CHAPTER 5

余元系列

5.1 Yuyuan-Bart-139M

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

5.1.1 简介 Brief Introduction

生物医疗领域的生成语言模型，英文的 BioBART-base。

A generative language model for biomedicine, BioBART-base in English.

5.1.2 模型分类 Model Taxonomy

5.1.3 模型信息 Model Information

Paper: BioBART: Pretraining and Evaluation of A Biomedical Generative Language Model

Yuyuan-Bart-139M 由清华大学和 IDEA 研究院一起提供的生物医疗领域的生成语言模型。我们使用 PubMed 上的生物医学研究论文摘要（约 41G）作为预训练语料。使用开源框架 DeepSpeed 的情况下，我们在 2 个带有 16 个 40GB A100 GPU 的 DGX 结点上对 BioBART-base（139M 参数）进行了约 100 小时的训练。

The Yuyuan-Bart-139M is a biomedical generative language model jointly produced by Tsinghua University and International Digital Economy Academy (IDEA). We use biomedical research paper abstracts on PubMed (41G) as the

pretraining corpora. We train the base version of BioBART(139M parameters) on 2 DGX with 16 40GB A100 GPUs for about 100 hours with the help of the open-resource framework DeepSpeed.

5.1.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Yuyuan-Bart-139M

加载模型 Loading Models

```
from transformers import BartForConditionalGeneration, BartTokenizer
tokenizer = BartTokenizer.from_pretrained('IDEA-CCNL/Yuyuan-Bart-139M')
model = BartForConditionalGeneration.from_pretrained('IDEA-CCNL/Yuyuan-Bart-139M')

text = 'Influenza is a <mask> disease.'
input_ids = tokenizer([text], return_tensors="pt")['input_ids']
model.eval()
generated_ids = model.generate(
    input_ids=input_ids,
)
preds = [tokenizer.decode(g, skip_special_tokens=True, clean_up_tokenization_
↪spaces=True) for g in generated_ids]
print(preds)
```

5.1.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的对该模型的论文：

If you are using the resource for your work, please cite the our paper for this model:

```
@misc{BioBART,
  title={BioBART: Pretraining and Evaluation of A Biomedical Generative Language␣
↪Model},
  author={Hongyi Yuan and Zheng Yuan and Ruyi Gan and Jiaxing Zhang and Yutao Xie and␣
↪Sheng Yu},
  year={2022},
  eprint={2204.03905},
  archivePrefix={arXiv}
}
```

如果您在您的工作中使用了我们的模型，也可以引用我们的总论文：

If you are using the resource for your work, please cite the our overview paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

5.2 Yuyuan-Bart-400M

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

5.2.1 简介 Brief Introduction

生物医疗领域的生成语言模型，英文的 BioBART-large。

A generative language model for biomedicine, BioBART-large in English.

5.2.2 模型分类 Model Taxonomy

5.2.3 模型信息 Model Information

Paper: BioBART: Pretraining and Evaluation of A Biomedical Generative Language Model

Yuyuan-Bart-139M 由清华大学和 IDEA 研究院一起提供的生物医疗领域的生成语言模型。我们使用 PubMed 上的生物医学研究论文摘要（约 41G）作为预训练语料。使用开源框架 DeepSpeed 的情况下，我们在 2 个带有 16 个 40GB A100 GPU 的 DGX 结点上对 BioBART-large（400M 参数）进行了约 168 小时的训练。

The Yuyuan-Bart-139M is a biomedical generative language model jointly produced by Tsinghua University and International Digital Economy Academy (IDEA). We use biomedical research paper abstracts on PubMed (41G) as the pretraining corpora. We train the base version of BioBART(139M parameters) on 2 DGX with 16 40GB A100 GPUs for about 168 hours with the help of the open-resource framework DeepSpeed.

5.2.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Yuyuan-Bart-400M

加载模型 Loading Models

```
from transformers import BartForConditionalGeneration, BartTokenizer
tokenizer = BartTokenizer.from_pretrained('IDEA-CCNL/Yuyuan-Bart-400M')
model = BartForConditionalGeneration.from_pretrained('IDEA-CCNL/Yuyuan-Bart-400M')

text = 'Influenza is a <mask> disease.'
input_ids = tokenizer([text], return_tensors="pt")['input_ids']
model.eval()
generated_ids = model.generate(
    input_ids=input_ids,
)
preds = [tokenizer.decode(g, skip_special_tokens=True, clean_up_tokenization_
↪spaces=True) for g in generated_ids]
print(preds)
```

5.2.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的对该模型的论文：

If you are using the resource for your work, please cite the our paper for this model:

```
@misc{BioBART,
  title={BioBART: Pretraining and Evaluation of A Biomedical Generative Language Model},
  author={Hongyi Yuan and Zheng Yuan and Ruyi Gan and Jiaxing Zhang and Yutao Xie and Sheng Yu},
  year={2022},
  eprint={2204.03905},
  archivePrefix={arXiv}
}
```

如果您在您的工作中使用了我们的模型，也可以引用我们的总论文：

If you are using the resource for your work, please cite the our overview paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

5.3 Yuyuan-GPT2-3.5B

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

5.3.1 简介 Brief Introduction

目前最大的，医疗领域的生成语言模型 GPT2。

The currently largest, generative language model GPT2 in the medical domain.

5.3.2 模型分类 Model Taxonomy

5.3.3 模型信息 Model Information

我们采用与 Wenzhong-GPT2-3.5B 相同的架构，在 50GB 的医学 (PubMed) 语料库上进行预训练。我们使用了 32 个 NVIDIA A100 显卡大约 7 天。我们的 Yuyuan-GPT2-3.5B 是医疗领域最大的开源的 GPT2 模型。进一步地，模型可以通过计算困惑度（PPL）来判断事实。为了完成问答功能，我们将短语模式从疑问的形式转换为了陈述句。

We adopt the same architecture as Wenzhong-GPT2-3.5B to be pre-trained on 50 GB medical (PubMed) corpus. We use 32 NVIDIA A100 GPUs for about 7 days. Our Yuyuan-GPT2-3.5B is the largest open-source GPT2 model in the medical domain. We further allow the model to judge facts by computing perplexity (PPL). To accomplish question-and-answer functionality, we transform the phrase pattern from interrogative to declarative.

5.3.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Yuyuan-GPT2-3.5B

加载模型 Loading Models

```
from transformers import GPT2Tokenizer, GPT2Model
tokenizer = GPT2Tokenizer.from_pretrained('IDEA-CCNL/Yuyuan-GPT2-3.5B')
model = GPT2Model.from_pretrained('IDEA-CCNL/Yuyuan-GPT2-3.5B')
text = "Replace me by any text you'd like."
encoded_input = tokenizer(text, return_tensors='pt')
output = model(**encoded_input)
```

使用示例 Usage Examples

```
from transformers import pipeline, set_seed
set_seed(55)
generator = pipeline('text-generation', model='IDEA-CCNL/Yuyuan-GPT2-3.5B')
generator("Diabetics should not eat", max_length=30, num_return_sequences=1)
```

5.3.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

5.4 YuyuanQA-GPT2-3.5B

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

5.4.1 简介 Brief Introduction

善于处理医疗问答任务，医疗的领域模型，英文版的 GPT2。

Good at handling medical question answering tasks, a medical domain model, GPT2 in English.

5.4.2 模型分类 Model Taxonomy

5.4.3 模型信息 Model Information

问答在自然语言处理领域中反映 AI 系统的知识水平的重要任务。为了可以在医疗领域中使用强大的问答能力的语言模型，我们基于 Yuyuan-GPT2-3.5B，对其使用了 10K 条医疗的问答对进行微调。我们希望探索一种简单、有效的方式直接实现问答系统而不需要额外的设计，即利用大模型强大的记忆力和理解能力。

Question answering (QA) is an important task in the Natural Language Processing to present the knowledge level of AI systems. To provide a language model with powerful QA capability in the medical domain, we fine-tuned Yuyuan-GPT2-3.5B on 10K medical Q&A pairs.

模型 Model

finetune 的模型是 yuyuan 模型，余元模型是 GPT2 的结构，在预训练阶段主要是用 PubMed 医疗相关的数据集进行的预训练。是一个医疗领域的大模型。模型共有 35 亿参数，主要参数如下表所示：

预训练的数据，主要医疗相关的论文、杂志期刊等，以英文语料为主。

数据 Data

用于 finetune 的语料是清洗于MedQuAD数据集，清洗完成后是下面的格式：

```
......
{'question':'.........','answer':'........'}
{'question':'.........','answer':'........'}
......
```

框架 Framework

finetune 的框架是 IDEA 研究院 CCNL 小组整合各大框架的优点开源的封神框架，具体代码详见：Fengshenbang-LM/fengshen/examples/wenzhong_qa/finetune_medicalQA.py 和 Fengshenbang-LM/fengshen/data/task_dataloader/medicalQADataset.py。

训练参数 Parameter

训练参数，我们采用了 deepspeed 相关的配置，用 2 个集群的节点共 16 张 A100，在很短的时间内完成了 finetune。具体参数配置可以参考：Fengshenbang-LM/fengshen/examples/wenzhong_qa/finetune_GPT2_medicalQA.sh

效果对比 Results

finetune 后的模型，用 100 对问答对，基于 BLEU 分与之前用 Megatron 框架训练的模型进行了简单的对比，效果比较接近。

unsmoth method:

smoth method:

下游任务 Performance

我们测试了该模型在未见过的 100 条 QA 对上的表现：

We tested the model on 100 unseen QA pairs:

5.4.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：YuyuanQA-GPT2-3.5B

加载模型 Loading Models

```
from transformers import GPT2Tokenizer,GPT2LMHeadModel

hf_model_path = 'YuyuanQA-GPT2-3.5B'

tokenizer = GPT2Tokenizer.from_pretrained(hf_model_path)
model = GPT2LMHeadModel.from_pretrained(hf_model_path)
```

使用示例 Usage Examples

```
fquestion = "What should gout patients pay attention to in diet?"
inputs = tokenizer(f'Question:{question} answer:',return_tensors='pt')

generation_output = model.generate(**inputs,
                                   return_dict_in_generate=True,
                                   output_scores=True,
                                   max_length=150,
                                   # max_new_tokens=80,
                                   do_sample=True,
                                   top_p = 0.6,
                                   eos_token_id=50256,
                                   pad_token_id=0,
                                   num_return_sequences = 5)

for idx,sentence in enumerate(generation_output.sequences):
    print('next sentence %d:\n'%idx,
          tokenizer.decode(sentence).split('<|endoftext|>')[0])
    print('*'*40)
```

回答问题 Answering the Questions

支持直接用 Haggingface 或者 pytorch-lightning 框架调用。由于在 finetune 的时候，加入了 prompt，在问答的时候，输入应该是："Question:your question about medical? answer:", 接着模型就回以续写的方式回答你的问题。用 huggingface 的调用代码可以参考下面的代码：

```
from transformers import GPT2Tokenizer,GPT2LMHeadModel
model_path = 'pretrained_model_hf/yuyuanQA-v1' # input your own model file path
model = GPT2LMHeadModel.from_pretrained(model_path)
tokenizer = GPT2Tokenizer.from_pretrained(model_path)
model = model.cuda(6) # move your model to the GPU
model.eval() # just do predict

def answering(question):
# question = "What should gout patients pay attention to in diet?"
    inputs = tokenizer(f'Question:{question} answer:',return_tensors='pt').input_ids.
↪to(model.device)

    generation_output = model.generate(input_ids = inputs,
                                   return_dict_in_generate=True,
                                   output_scores=True,
                                   max_length=150,
```

(续下页)

(接上页)

```
                                # max_new_tokens=80,
                                do_sample=True,
                                top_p = 0.9,
                                eos_token_id=50256,
                                pad_token_id=0,
                                num_return_sequences = 5)
    answers = []
    for idx,sentence in enumerate(generation_output.sequences):
        next_sentence = tokenizer.decode(sentence).split('<|endoftext|>')[0]
        answer = next_sentence.split(sep='answer:',maxsplit=1)[1]
        answers.append(answer)
    return answers
answering('your question?')
```

演示 Demo

我们用该模型做了一个医疗问答演示。将来，我们会将这款产品做成微信小程序与大家见面。

We made a demo of medical QA system with this model. In the future, we will make this product into a wechat app to meet you.

Demo for MedicalQA

请输入你的问题:

痛风患者在饮食中应该注意什么?

提交

你的问题是: What should gout patients pay attention to in their diet?

候选回答「1」:

中文：嘌呤是许多食物中的物质，饮食中嘌呤含量低会导致痛风。嘌呤在肾脏被分解为尿酸，血液中高水平的尿酸会导致痛风。许多食物中都含有高蛋白质和钠，这种饮食也会导致痛风。

英文： A diet low in purines, which are substances found in many foods, can cause gout.Purines are broken down to uric acid in the kidneys, and high levels of uric acid in the blood can cause gout.A diet high in protein and sodium, which are found in many foods, can also cause gout.

候选回答「2」:

中文：痛风患者应尽量限制嘌呤的摄入，嘌呤存在于海鲜、贝类和啤酒等食物中。当身体分解食物中的嘌呤时，血液中会积累嘌呤，尤其是在饮食不受控制的情况下。嘌呤如果不能被分解，也会在关节中积聚。

英文： Gout patients should try to limit their intake of purines, which are found in foods such as seafood, shellfish, and beer.Purines can build up in the blood when the body breaks down purines that are found in food, especially if the diet is not controlled.Purines can also build up in the joints if they cannot be broken down.

候选回答「3」:

中文：痛风患者应限制钠（盐）的摄入，钠存在于盐和许多加工食品中。然而，一些加工食品的钠含量很高。这包括加工肉类、烘焙食品、咖啡奶油和一些调味品。痛风患者应限制磷的摄入，许多加工食品中都含有磷。有些加工食品含磷量很高。

5.4.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

CHAPTER 6

周文王系列

6.1 Zhouwenwang-Unified-1.3B

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

6.1.1 简介 Brief Introduction

与追一科技合作探索的中文统一模型，13 亿参数的编码器结构模型。

The Chinese unified model explored in cooperation with Zhuiyi Technology, the encoder structure model with 1.3B parameters.

6.1.2 模型分类 Model Taxonomy

6.1.3 模型信息 Model Information

IDEA 研究院认知计算中心联合追一科技有限公司提出的具有新结构的大模型。该模型在预训练阶段时考虑统一 LM 和 MLM 的任务，这让其同时具备生成和理解的能力，并且增加了旋转位置编码技术。目前已有 13 亿参数的 Zhouwenwang-Unified-1.3B 大模型，是中文领域中可以同时做 LM 和 MLM 任务的最大的模型。我们后续会持续在模型规模、知识融入、监督辅助任务等方向不断优化。

A large-scale model (Zhouwenwang-Unified-1.3B) with a new structure proposed by IDEA CCNL and Zhuiyi Technology. The model considers the task of unifying LM (Language Modeling) and MLM (Masked Language Modeling) during

the pre-training phase, which gives it both generative and comprehension capabilities, and applys rotational position encoding. At present, Zhouwenwang-Unified-1.3B with 13B parameters is the largest Chinese model that can do both LM and MLM tasks. In the future, we will continue to optimize it in the direction of model size, knowledge incorporation, and supervisory assistance tasks.

下游任务 Performance

下游中文任务的得分（没有做任何数据增强）。

Scores on downstream chinese tasks (without any data augmentation)

6.1.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Zhouwenwang-Unified-1.3B

加载模型 Loading Models

因为transformers库中是没有 Zhouwenwang-Unified-1.3B 相关的模型结构的，所以你可以在我们的Fengshenbang-LM中找到并且运行代码。

Since there is no structure of Zhouwenwang-Unified-1.3B in transformers library, you can find the structure of Zhouwenwang-Unified-1.3B and run the codes in Fengshenbang-LM.

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
```

```
from fengshen import RoFormerModel
from fengshen import RoFormerConfig
from transformers import BertTokenizer

tokenizer = BertTokenizer.from_pretrained("IDEA-CCNL/Zhouwenwang-Unified-1.3B")
config = RoFormerConfig.from_pretrained("IDEA-CCNL/Zhouwenwang-Unified-1.3B")
model = RoFormerModel.from_pretrained("IDEA-CCNL/Zhouwenwang-Unified-1.3B")
```

使用示例 Usage Examples

你可以使用该模型进行续写任务。

You can use the model for continuation writing tasks.

```
from fengshen import RoFormerModel
from transformers import AutoTokenizer
import torch
import numpy as np

sentence = '清华大学位于'
max_length = 32

tokenizer = AutoTokenizer.from_pretrained("IDEA-CCNL/Zhouwenwang-Unified-1.3B")
model = RoFormerModel.from_pretrained("IDEA-CCNL/Zhouwenwang-Unified-1.3B")

for i in range(max_length):
    encode = torch.tensor(
        [[tokenizer.cls_token_id]+tokenizer.encode(sentence, add_special_
↪tokens=False)]).long()
    logits = model(encode)[0]
    logits = torch.nn.functional.linear(
        logits, model.embeddings.word_embeddings.weight)
    logits = torch.nn.functional.softmax(
        logits, dim=-1).cpu().detach().numpy()[0]
    sentence = sentence + \
        tokenizer.decode(int(np.random.choice(logits.shape[1], p=logits[-1])))
    if sentence[-1] == '。':
        break
print(sentence)
```

6.1.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

---

language:

- zh license: apache-2.0 widget:
- text: " 生活的真谛是 [MASK]。"

6.2 Zhouwenwang-Unified-110M

- Github: Fengshenbang-LM
- Docs: Fengshenbang-Docs

6.2.1 简介 Brief Introduction

与追一科技合作探索的中文统一模型，1.1 亿参数的编码器结构模型。

The Chinese unified model explored in cooperation with Zhuiyi Technology, the encoder structure model with 110M parameters.

6.2.2 模型分类 Model Taxonomy

6.2.3 模型信息 Model Information

IDEA 研究院认知计算中心联合追一科技有限公司提出的具有新结构的大模型。该模型在预训练阶段时考虑统一 LM 和 MLM 的任务，这让其同时具备生成和理解的能力，并且增加了旋转位置编码技术。我们后续会持续在模型规模、知识融入、监督辅助任务等方向不断优化。

A large-scale model (Zhouwenwang-Unified-1.3B) with a new structure proposed by IDEA CCNL and Zhuiyi Technology. The model considers the task of unifying LM (Language Modeling) and MLM (Masked Language Modeling) during the pre-training phase, which gives it both generative and comprehension capabilities, and applys rotational position encoding. In the future, we will continue to optimize it in the direction of model size, knowledge incorporation, and supervisory assistance tasks.

6.2.4 使用 Usage

模型下载地址 Download Address

Huggingface 地址：Zhouwenwang-Unified-110M

加载模型 Loading Models

因为transformers库中是没有 Zhouwenwang-Unified-110M 相关的模型结构的，所以你可以在我们的Fengshenbang-LM中找到并且运行代码。

Since there is no structure of Zhouwenwang-Unified-110M in transformers library, you can find the structure of Zhouwenwang-Unified-110M and run the codes in Fengshenbang-LM.

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
```

```
from fengshen import RoFormerModel
from fengshen import RoFormerConfig
from transformers import BertTokenizer

tokenizer = BertTokenizer.from_pretrained("IDEA-CCNL/Zhouwenwang-Unified-110M")
config = RoFormerConfig.from_pretrained("IDEA-CCNL/Zhouwenwang-Unified-110M")
model = RoFormerModel.from_pretrained("IDEA-CCNL/Zhouwenwang-Unified-110M")
```

使用示例 Usage Examples

你可以使用该模型进行续写任务。

You can use the model for continuation writing tasks.

```
from fengshen import RoFormerModel
from transformers import AutoTokenizer
import torch
import numpy as np

sentence = '清华大学位于'
max_length = 32

tokenizer = AutoTokenizer.from_pretrained("IDEA-CCNL/Zhouwenwang-Unified-110M")
model = RoFormerModel.from_pretrained("IDEA-CCNL/Zhouwenwang-Unified-110M")

for i in range(max_length):
    encode = torch.tensor(
        [[tokenizer.cls_token_id]+tokenizer.encode(sentence, add_special_
```

(续下页)

(接上页)

```
↪tokens=False)]).long()
    logits = model(encode)[0]
    logits = torch.nn.functional.linear(
        logits, model.embeddings.word_embeddings.weight)
    logits = torch.nn.functional.softmax(
        logits, dim=-1).cpu().detach().numpy()[0]
    sentence = sentence + \
        tokenizer.decode(int(np.random.choice(logits.shape[1], p=logits[-1])))
    if sentence[-1] == '。':
        break
print(sentence)
```

6.2.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

CHAPTER 7

场景应用

7.1 文本情感分类 Sentiment Analysis

分析处理带有情感色彩的主观性文本，判断文本观点的正负倾向性。Analyze and handle subjective texts with emotional overtones, and determine the positive and negative tendencies of textual views.

7.1.1 推荐模型 Recommended Models

Erlangshen-Roberta-110M-Sentiment：RoBERTa-wwm-ext-base 在数个情感分析任务微调后的版本。

Erlangshen-Roberta-330M-Sentiment：RoBERTa-wwm-ext-large 在数个情感分析任务微调后的版本。

Erlangshen-MegatronBert-1.3B-Sentiment：2021 年登顶 FewCLUE 和 ZeroCLUE 的中文 BERT，在数个情感分析任务微调后的版本。

7.1.2 下游效果 Performance

| 模型 Model | ASAP-SENT | ASAP-ASPECT | ChnSentiCorp | | :--------: | :-----: | :----: | :-----: | | Erlangshen-Roberta-110M-Sentiment | 97.77 | 97.31 | 96.61 | | Erlangshen-Roberta-330M-Sentiment | 97.9 | 97.51 | 96.66 || Erlangshen-MegatronBert-1.3B-Sentiment | 98.1 | 97.8 | 97 |

7.1.3 使用 Usage

```
from transformers import AutoModelForSequenceClassification
from transformers import BertTokenizer
import torch

tokenizer=BertTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-110M-Sentiment')
model=AutoModelForSequenceClassification.from_pretrained('IDEA-CCNL/Erlangshen-
↪Roberta-110M-Sentiment')

text='今天心情不好'

output=model(torch.tensor([tokenizer.encode(text)]))
print(torch.nn.functional.softmax(output.logits,dim=-1))
```

7.1.4 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

7.2 自然语言推理 Natural Language Inference

判断句子对之间的语义逻辑关系。Determine whether a “hypothesis” is true (entailment), false (contradiction), or undetermined (neutral) given a “premise”.

7.2.1 推荐模型 Recommended Models

零样本/少样本 Zero-shot/Few-shot

Erlangshen-UniMC-RoBERTa-110M-Chinese：基于chinese-roberta-wwm-ext，将自然语言理解任务转化为多选任务，并且使用多个 NLU 任务来进行预训练。

Erlangshen-UniMC-RoBERTa-330M-Chinese：基于chinese-roberta-wwm-ext-large，将自然语言理解任务转化为多选任务，并且使用多个 NLU 任务来进行预训练。

Erlangshen-UniMC-DeBERTa-v2-110M-Chinese：基于Erlangshen-DeBERTa-v2-97M-Chinese，将自然语言理解任务转化为多选任务，并且使用多个 NLU 任务来进行预训练。

Erlangshen-UniMC-DeBERTa-v2-330M-Chinese：基于Erlangshen-DeBERTa-v2-320M-Chinese，将自然语言理解任务转化为多选任务，并且使用多个 NLU 任务来进行预训练。

Erlangshen-MacBERT-325M-NLI-Chinese：3.25 亿参数的 MacBERT，在 NLI 任务上进行预训练，并在 FewCLUE 的 OCNLI 任务上微调。

Erlangshen-UniMC-MegatronBERT-1.3B-Chinese：基于Erlangshen-MegatronBert-1.3B，将自然语言理解任务转化为多选任务，并且使用多个 NLU 任务来进行预训练。

微调 Fine-tuning

Erlangshen-Roberta-110M-NLI：中文的chinese-roberta-wwm-ext在数个推理任务微调后的版本。

Erlangshen-Roberta-330M-NLI：中文的chinese-roberta-wwm-ext-large在数个推理任务微调后的版本。

Erlangshen-MegatronBert-1.3B-NLI：2021 年登顶 FewCLUE 和 ZeroCLUE 的中文 BERT，在数个推理任务微调后的版本。

7.2.2 下游效果 Performance

零样本 Zero-shot

少样本 Few-shot

微调 Fine-tuning

7.2.3 使用 Usage

UniMC / MacBERT

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
cd Fengshenbang-LM
pip install --editable .
```

```
import argparse
from fengshen.pipelines.multiplechoice import UniMCPipelines


total_parser = argparse.ArgumentParser("TASK NAME")
total_parser = UniMCPipelines.piplines_args(total_parser)
args = total_parser.parse_args()
pretrained_model_path = 'IDEA-CCNL/Erlangshen-UniMC-RoBERTa-110M-Chinese'
args.learning_rate=2e-5
args.max_length=512
args.max_epochs=3
args.batchsize=8
args.default_root_dir='./'
model = UniMCPipelines(args, pretrained_model_path)

train_data = []
dev_data = []
test_data = [
        {"texta": "要稳定和完善出口政策,加快通关便利化改革,扩大跨境电子商务试点",
         "textb": "",
         "question": "基于文本",
         "choice": [
            "可以推出：外来货物入境不需要经过海关",
            "不能推出：外来货物入境不需要经过海关",
            "很难推出：外来货物入境不需要经过海关"
            ],
         "answer": "不能推出：外来货物入境不需要经过海关",
```

(续下页)

(接上页)

```
        "label": 1,
        "id": 23}
    ]

if args.train:
    model.train(train_data, dev_data)
result = model.predict(test_data)
for line in result[:20]:
    print(line)
```

其他 Other

```
from transformers import BertForSequenceClassification
from transformers import BertTokenizer
import torch

tokenizer=BertTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-110M-NLI')
model=BertForSequenceClassification.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-
↪110M-NLI')

texta='今天的饭不好吃'
textb='今天心情不好'

output=model(torch.tensor([tokenizer.encode(texta,textb)]))
print(torch.nn.functional.softmax(output.logits,dim=-1))
```

7.2.4 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
```

(续下页)

(接上页)

```
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

7.3 文本相似度 Text Similarity

计算文本之间的语义相似度。Calculate the semantic similarity between texts.

7.3.1 推荐模型 Recommended Models

Erlangshen-Roberta-110M-Similarity：RoBERTa-wwm-ext-base 在数个相似度任务微调后的版本。

Erlangshen-Roberta-330M-Similarity：RoBERTa-wwm-ext-large 在数个相似度任务微调后的版本。

Erlangshen-MegatronBert-1.3B-Similarity：2021 年登顶 FewCLUE 和 ZeroCLUE 的中文 BERT，在数个情感分析任务微调后的版本。

7.3.2 下游效果 Performance

| Model | BQ | BUSTM | AFQMC | | :--------: | :-----: | :----: | :-----: | | Erlangshen-Roberta-110M-Similarity | 85.41 | 95.18 | 81.72 | | Erlangshen-Roberta-330M-Similarity | 86.21 | 99.29 | 93.89 || Erlangshen-MegatronBert-1.3B-Similarity | 86.31 | - | - |

7.3.3 使用 Usage

```
from transformers import BertForSequenceClassification
from transformers import BertTokenizer
import torch

tokenizer=BertTokenizer.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-110M-Similarity
```

(续下页)

(接上页)

```
↪')
model=BertForSequenceClassification.from_pretrained('IDEA-CCNL/Erlangshen-Roberta-
↪110M-Similarity')

texta='今天的饭不好吃'
textb='今天心情不好'

output=model(torch.tensor([tokenizer.encode(texta,textb)]))
print(torch.nn.functional.softmax(output.logits,dim=-1))
```

7.3.4 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

7.4 关系抽取 Sentiment Analysis

在文本数据中抽取特定实体及实体之间存在的关系。Extract specific entities and relationships between entities in text data.

7.4.1 推荐模型 Recommended Models

Erlangshen-Ubert-110M-Chinese：采用统一的框架处理多种抽取任务，AIWIN2022 的冠军方案，1.1 亿参数量的中文 UBERT-Base。

Erlangshen-Ubert-330M-Chinese：采用统一的框架处理多种抽取任务，AIWIN2022 的冠军方案，1.1 亿参数量的中文 UBERT-Large。

7.4.2 使用 Usage

Pip install fengshen:

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
cd Fengshenbang-LM
pip install --editable ./
```

Run the code:

```
import argparse
from fengshen import UbertPiplines

total_parser = argparse.ArgumentParser("TASK NAME")
total_parser = UbertPiplines.piplines_args(total_parser)
args = total_parser.parse_args()

args.pretrained_model_path = "IDEA-CCNL/Erlangshen-Ubert-110M-Chinese"

test_data=[
    {
        "task_type": "抽取任务",
        "subtask_type": "实体识别",
        "text": "这也让很多业主据此认为，雅清苑是政府公务员挤对了国家的经适房政策。",
        "choices": [
            {"entity_type": "小区名字"},
            {"entity_type": "岗位职责"}
            ],
        "id": 0}
]
```

(续下页)

(接上页)

```
model = UbertPiplines(args)
result = model.predict(test_data)
for line in result:
    print(line)
```

7.4.3 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

7.5 事件抽取 Event Extraction

在文本数据中抽取特定事件信息如事件发生的事件、地点、人物等。Extract specific event information such as event, location, person, etc. from text data.

7.5.1 推荐模型 Recommended Models

Erlangshen-Ubert-110M-Chinese：采用统一的框架处理多种抽取任务，AIWIN2022 的冠军方案，1.1 亿参数量的中文 UBERT-Base。

Erlangshen-Ubert-330M-Chinese：采用统一的框架处理多种抽取任务，AIWIN2022 的冠军方案，1.1 亿参数量的中文 UBERT-Large。

7.5.2 使用 Usage

Pip install fengshen:

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
cd Fengshenbang-LM
pip install --editable ./
```

Run the code:

```
import argparse
from fengshen import UbertPiplines

total_parser = argparse.ArgumentParser("TASK NAME")
total_parser = UbertPiplines.piplines_args(total_parser)
args = total_parser.parse_args()

args.pretrained_model_path = "IDEA-CCNL/Erlangshen-Ubert-110M-Chinese"

test_data=[
    {
        "task_type": "抽取任务",
        "subtask_type": "实体识别",
        "text": "这也让很多业主据此认为，雅清苑是政府公务员挤对了国家的经适房政策。",
        "choices": [
            {"entity_type": "小区名字"},
            {"entity_type": "岗位职责"}
            ],
        "id": 0}
]
```

(续下页)

(接上页)

```
model = UbertPiplines(args)
result = model.predict(test_data)
for line in result:
    print(line)
```

7.5.3 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

7.6 阅读理解 Reading Comprehension

对指定文本数据内容进行理解和分析并回答提出的问题。Understand and analyze the content of specified text data and answer the questions.

7.6.1 推荐模型 Recommended Models

Erlangshen-Ubert-110M-Chinese：采用统一的框架处理多种抽取任务，AIWIN2022 的冠军方案，1.1 亿参数量的中文 UBERT-Base。

Erlangshen-Ubert-330M-Chinese：采用统一的框架处理多种抽取任务，AIWIN2022 的冠军方案，1.1 亿参数量的中文 UBERT-Large。

7.6.2 使用 Usage

Pip install fengshen:

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
cd Fengshenbang-LM
pip install --editable ./
```

Run the code:

```
import argparse
from fengshen import UbertPiplines

total_parser = argparse.ArgumentParser("TASK NAME")
total_parser = UbertPiplines.piplines_args(total_parser)
args = total_parser.parse_args()

args.pretrained_model_path = "IDEA-CCNL/Erlangshen-Ubert-110M-Chinese"

test_data=[
    {
        "task_type": "抽取任务",
        "subtask_type": "实体识别",
        "text": "这也让很多业主据此认为，雅清苑是政府公务员挤对了国家的经适房政策。",
        "choices": [
            {"entity_type": "小区名字"},
            {"entity_type": "岗位职责"}
            ],
        "id": 0}
]
```

(续下页)

(接上页)

```
model = UbertPiplines(args)
result = model.predict(test_data)
for line in result:
    print(line)
```

7.6.3 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

7.7 实体识别 Named-entity recognition

从文本数据中获取人名、地名等实体数据。Obtain entity data such as person name, place name, etc. from text data.

7.7.1 推荐模型 Recommended Models

Erlangshen-Ubert-110M-Chinese：采用统一的框架处理多种抽取任务，AIWIN2022 的冠军方案，1.1 亿参数量的中文 UBERT-Base。

Erlangshen-Ubert-330M-Chinese：采用统一的框架处理多种抽取任务，AIWIN2022 的冠军方案，1.1 亿参数量的中文 UBERT-Large。

7.7.2 使用 Usage

Pip install fengshen:

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
cd Fengshenbang-LM
pip install --editable ./
```

Run the code:

```
import argparse
from fengshen import UbertPiplines

total_parser = argparse.ArgumentParser("TASK NAME")
total_parser = UbertPiplines.piplines_args(total_parser)
args = total_parser.parse_args()

args.pretrained_model_path = "IDEA-CCNL/Erlangshen-Ubert-110M-Chinese"

test_data=[
    {
        "task_type": "抽取任务",
        "subtask_type": "实体识别",
        "text": "这也让很多业主据此认为，雅清苑是政府公务员挤对了国家的经适房政策。",
        "choices": [
            {"entity_type": "小区名字"},
            {"entity_type": "岗位职责"}
            ],
        "id": 0}
]
```

(续下页)

(接上页)

```
model = UbertPiplines(args)
result = model.predict(test_data)
for line in result:
    print(line)
```

7.7.3 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

7.8 文本生成图片 Text-to-Image Generation

根据文本的描述创建相关图像。Create relevant images based on the given text.

7.8.1 推荐模型 Recommended Models

Taiyi-Stable-Diffusion-1B-Chinese-v0.1：首个开源的中文 Stable Diffusion 模型，基于 0.2 亿筛选过的中文图文对训练。

Taiyi-Stable-Diffusion-1B-Chinese-EN-v0.1：首个开源的中英双语 Stable Diffusion 模型，基于 0.2 亿筛选过的中文图文对训练。

Taiyi-Diffusion-532M-Nature-Chinese：由Katherine Crowson's的无条件扩散模型在 1k+ 张收集的自然风景图上微调而来。结合IDEA-CCNL/Taiyi-CLIP-Roberta-large-326M-Chinese可以实现中文 Guided Diffusion 的生成方式。

Taiyi-Diffusion-532M-Cyberpunk-Chinese：由Katherine Crowson's的无条件扩散模型在 1k+ 张收集的赛博朋克风的图上微调而来。结合IDEA-CCNL/Taiyi-CLIP-Roberta-large-326M-Chinese可以实现中文 Guided Diffusion 的生成方式。

7.8.2 使用 Usage

Stable-Diffusion

全精度 Full precision

```
from diffusers import StableDiffusionPipeline

pipe = StableDiffusionPipeline.from_pretrained("IDEA-CCNL/Taiyi-Stable-Diffusion-1B-
↪Chinese-v0.1").to("cuda")

prompt = '飞流直下三千尺，油画'
image = pipe(prompt, guidance_scale=7.5).images[0]
image.save("飞流.png")
```

半精度 Half precision FP16 (CUDA)

添加 `torch_dtype=torch.float16` 和 `device_map="auto"` 可以快速加载 FP16 的权重，以加快推理速度。更多信息见 the optimization docs。

```
# !pip install git+https://github.com/huggingface/accelerate
import torch
from diffusers import StableDiffusionPipeline
torch.backends.cudnn.benchmark = True
pipe = StableDiffusionPipeline.from_pretrained("IDEA-CCNL/Taiyi-Stable-Diffusion-1B-
↪Chinese-v0.1", torch_dtype=torch.float16)
pipe.to('cuda')
```

(续下页)

(接上页)

```
prompt = '飞流直下三千尺，油画'
image = pipe(prompt, guidance_scale=7.5).images[0]
image.save("飞流.png")
```

Diffusion

使用示例见：https://github.com/IDEA-CCNL/Fengshenbang-LM/tree/main/fengshen/examples/disco_project

7.8.3 生成示例 Example

Taiyi-Stable-Diffusion-1B-Chinese-v0.1Taiyi-Stable-Diffusion-1B-Chinese-EN-v0.1Taiyi-Diffusion-532M-Nature-ChineseTaiyi-Diffusion-532M-Cyberpunk-Chinese

7.8.4 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

7.9 医疗问答 Medical Question Answering

通过医疗问答对 Finetune 完成闭卷问答（Closed-book QA）任务。Complete Closed-book QA tasks by fine-tuning on Medical Q&A pairs.

7.9.1 推荐模型 Recommended Models

YuyuanQA-GPT2-3.5B：善于处理医疗问答任务，医疗的领域模型，英文版的 GPT2。

7.9.2 下游效果 Performance

我们测试了该模型在未见过的 100 条 QA 对上的表现：

We tested the model on 100 unseen QA pairs:

7.9.3 使用 Usage

模型下载地址 Download Address

Huggingface 地址：YuyuanQA-GPT2-3.5B

加载模型 Loading Models

```
from transformers import GPT2Tokenizer,GPT2LMHeadModel

hf_model_path = 'YuyuanQA-GPT2-3.5B'

tokenizer = GPT2Tokenizer.from_pretrained(hf_model_path)
model = GPT2LMHeadModel.from_pretrained(hf_model_path)
```

使用示例 Usage Examples

```
fquestion = "What should gout patients pay attention to in diet?"
inputs = tokenizer(f'Question:{question} answer:',return_tensors='pt')

generation_output = model.generate(**inputs,
                                   return_dict_in_generate=True,
                                   output_scores=True,
                                   max_length=150,
                                   # max_new_tokens=80,
```

(续下页)

(接上页)

```
                                do_sample=True,
                                top_p = 0.6,
                                eos_token_id=50256,
                                pad_token_id=0,
                                num_return_sequences = 5)

for idx,sentence in enumerate(generation_output.sequences):
    print('next sentence %d:\n'%idx,
          tokenizer.decode(sentence).split('<|endoftext|>')[0])
    print('*'*40)
```

回答问题 Answering the Questions

支持直接用 Haggingface 或者 pytorch-lightning 框架调用。由于在 finetune 的时候，加入了 prompt，在问答的时候，输入应该是："Question:your question about medical? answer:"，接着模型就回以续写的方式回答你的问题。用 huggingface 的调用代码可以参考下面的代码：

```
from transformers import GPT2Tokenizer,GPT2LMHeadModel
model_path = 'pretrained_model_hf/yuyuanQA-v1' # input your own model file path
model = GPT2LMHeadModel.from_pretrained(model_path)
tokenizer = GPT2Tokenizer.from_pretrained(model_path)
model = model.cuda(6) # move your model to the GPU
model.eval() # just do predict

def answering(question):
# question = "What should gout patients pay attention to in diet?"
    inputs = tokenizer(f'Question:{question} answer:',return_tensors='pt').input_ids.
↪to(model.device)

    generation_output = model.generate(input_ids = inputs,
                                return_dict_in_generate=True,
                                output_scores=True,
                                max_length=150,
                                # max_new_tokens=80,
                                do_sample=True,
                                top_p = 0.9,
                                eos_token_id=50256,
                                pad_token_id=0,
                                num_return_sequences = 5)
    answers = []
    for idx,sentence in enumerate(generation_output.sequences):
        next_sentence = tokenizer.decode(sentence).split('<|endoftext|>')[0]
```

(续下页)

(接上页)

```
        answer = next_sentence.split(sep='answer:',maxsplit=1)[1]
        answers.append(answer)
    return answers
answering('your question?')
```

7.9.4 演示 Demo

我们用该模型做了一个医疗问答演示。将来，我们会将这款产品做成微信小程序与大家见面。

We made a demo of medical QA system with this model. In the future, we will make this product into a wechat app to meet you.

Demo for MedicalQA

请输入你的问题:

痛风患者在饮食中应该注意什么?

提交

你的问题是: What should gout patients pay attention to in their diet?

候选回答「1」:

中文：嘌呤是许多食物中的物质，饮食中嘌呤含量低会导致痛风。嘌呤在肾脏被分解为尿酸，血液中高水平的尿酸会导致痛风。许多食物中都含有高蛋白质和钠，这种饮食也会导致痛风。

英文： A diet low in purines, which are substances found in many foods, can cause gout.Purines are broken down to uric acid in the kidneys, and high levels of uric acid in the blood can cause gout.A diet high in protein and sodium, which are found in many foods, can also cause gout.

候选回答「2」:

中文：痛风患者应尽量限制嘌呤的摄入，嘌呤存在于海鲜、贝类和啤酒等食物中。当身体分解食物中的嘌呤时，血液中会积累嘌呤，尤其是在饮食不受控制的情况下。嘌呤如果不能被分解，也会在关节中积聚。

英文： Gout patients should try to limit their intake of purines, which are found in foods such as seafood, shellfish, and beer.Purines can build up in the blood when the body breaks down purines that are found in food, especially if the diet is not controlled.Purines can also build up in the joints if they cannot be broken down.

候选回答「3」:

中文：痛风患者应限制钠（盐）的摄入，钠存在于盐和许多加工食品中。然而，一些加工食品的钠含量很高。这包括加工肉类、烘焙食品、咖啡奶油和一些调味品。痛风患者应限制磷的摄入，许多加工食品中都含有磷。有些加工食品含磷量很高。

7.9.5 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文：

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站：

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

7.10 语义纠错 Semantic Denoising

对文中的字词、标点、语法等多类问题进行纠错，并针对性返回正确修改意见。Correct errors in wording, punctuation, grammar and many other types of problems in the text, and return correct corrections to the target.

7.10.1 推荐模型 Recommended Models

Randeng-Transformer-1.1B-Denoise：以语法纠错任务为微调目标的中文 Transformer-XL。

7.10.2 使用 Usage

加载模型 Loading Models

```
git clone https://github.com/IDEA-CCNL/Fengshenbang-LM.git
```

```
from fengshen.models.transfo_xl_denoise.tokenization_transfo_xl_denoise import␣
↪TransfoXLDenoiseTokenizer
from fengshen.models.transfo_xl_denoise.modeling_transfo_xl_denoise import␣
↪TransfoXLDenoiseModel

tokenizer = TransfoXLDenoiseTokenizer.from_pretrained('IDEA-CCNL/Randeng-Transformer-
↪1.1B-Denoise')
model = TransfoXLDenoiseModel.from_pretrained('IDEA-CCNL/Randeng-Transformer-1.1B-
↪Denoise')
```

使用示例 Usage Examples

```
from fengshen.models.transfo_xl_denoise.generate import denoise_generate
input_text = "凡是有成就的人，都很严肃地对待生命自己的"
res = denoise_generate(model, tokenizer,  input_text)
print(res)
# "有成就的人都很严肃地对待自己的生命。"
```

7.10.3 引用 Citation

如果您在您的工作中使用了我们的模型，可以引用我们的论文:

If you are using the resource for your work, please cite the our paper:

```
@article{fengshenbang,
  author    = {Junjie Wang and Yuxiang Zhang and Lin Zhang and Ping Yang and Xinyu␣
↪Gao and Ziwei Wu and Xiaoqun Dong and Junqing He and Jianheng Zhuo and Qi Yang and␣
↪Yongfeng Huang and Xiayu Li and Yanghan Wu and Junyu Lu and Xinyu Zhu and Weifeng␣
↪Chen and Ting Han and Kunhao Pan and Rui Wang and Hao Wang and Xiaojun Wu and␣
↪Zhongshen Zeng and Chongpei Chen and Ruyi Gan and Jiaxing Zhang},
  title     = {Fengshenbang 1.0: Being the Foundation of Chinese Cognitive␣
↪Intelligence},
  journal   = {CoRR},
  volume    = {abs/2209.02970},
  year      = {2022}
}
```

也可以引用我们的网站:

You can also cite our website:

```
@misc{Fengshenbang-LM,
  title={Fengshenbang-LM},
  author={IDEA-CCNL},
  year={2021},
  howpublished={\url{https://github.com/IDEA-CCNL/Fengshenbang-LM}},
}
```

CHAPTER 8

数据集列表

8.1 AFQMC 蚂蚁金融语义相似度

数据集介绍 https://www.cluebenchmarks.com/introduce.html

8.2 LSCTC 中文文本摘要

数据集介绍 https://chinesenlp.xyz/zh/docs/text_summarization.html

8.3 NLI 阅读理解合集

暂未开源

8.4 Sentiment 情感分析合集

暂未开源

8.5 Similarity 文本相似度合集

暂未开源

8.6 WuDao_180G 悟道开源预训练数据集

数据集介绍 https://www.sciencedirect.com/science/article/pii/S2666651021000152

上述列表为封神榜模型开源计划中所有用到的数据集，并且包含一部分数据处理预逻辑，全量原始数据暂时未开源，我们提供一部分数据 DEMO 以便使用者了解数据集的结构以及如何做数据处理，敬请期待后续开源计划。

DEMO 数据样式可以在 [github](https://github.com/IDEA-CCNL/fs_datasets/blob/master/raw_data/demo_data.jsonl) 查看。

CHAPTER 9

封神框架

9.1 参数管理

9.1.1 简介 Brief Introduction

框架中一些常用的参数说明文档。

9.1.2 数据相关参数

UniversalDataModule

该模块是通用的 DataModule，通常用于 fs_datasets 下的数据集，或者数据处理逻辑能用一个函数描述的 datasets，都用直接使用该 DataModule。datamodule.datasets 是一个 dict，结构类似于:

```
{
    "train":datasets,
    "validation":datasets,
    "test":datasets
}
```

- **num_workers**:fs_datasets 加载数据时的进程数，可以根据 CPU 的数目配置
- **dataloader_workers**:dataloader 处理数据的进程数，这个配置通常 2-4 就足够了，配置过大反而会导致处理数据效率降低甚至卡死的情况

- **train_batchsize**: 训练 batchsize
- **val_batchsize**: 验证 batchsize
- **test_batchsize**: 测试 batchsize
- **datasets_name**:fs_datasets 的名字，如果不传入的话则不会加载数据，需要用户显示指定 datamodule.datasets=xxxxx
- **train_datasets_field**:self.datasets 中训练集所对应的 key
- **val_datasets_field**:self.datasets 中验证集所对应的 key（有时 val 数据集会取用 test 集，所以设定了这三个参数做兼容
- **test_datasets_field**:self.datasets 中测试集所对应的 key
- **sampler_type**: 封神框架中自建的 sampler，用于支持大数据集

9.1.3 模型相关参数

add_module_args 通用模型参数

这个函数都是包括了一些常用的、跟特定模型无关的参数。

- **learning_rate**: 学习率
- **weight_decay**: 权重衰减
- **warmup_ratio**: 学习率 warmup 比例，比如设定是 0.1，总步数是 100，则前 10 步会 warmup 到最大值，并开始衰减
- **warmup_steps**: 学习率 warmup 步数，优先级大于 warmup_ratio
- **adam_beta1**:adam 参数
- **adam_beta2**:adam 参数
- **adam_epsilon**:adam 参数
- **model_path**: 模型路径
- **scheduler_type**: 支持多种 scheduler，包括 ['linear', 'cosine', 'cosine_with_restarts', 'polynomial', 'constant', 'constant_with_warmup']

9.1.4 训练相关参数

Lightning Trainer

Lightning Trainer 参数可以参考文档Doc

这里列举一些比较常用的:

- **max_epochs**: 设定总共 epoch 数
- **max_steps**: 设定总共的 steps 数，跟 max_epochs 冲突
- **gpus**: 机器的 gpu 数量
- **num_nodes**: 机器数
- **strategy**: 分布式策略，常用的比如 ddp, deepspeed_zero_stage_1，所有支持的可以参考 lightning 文档
- **gradient_clip_val**: 梯度裁剪
- **check_val_every_n_epoch**: 多少个 epoch 后做一次 validation
- **val_check_interval**: 在 epoch 内做 validation 的频率，如果是 float 则是按比例，如果是 int 型则是按 steps 算
- **precision**: 模型精度
- **default_root_dir**: 设定日志存放的目录

为了让大家好用封神榜大模型，参与大模型的继续训练和下游应用，我们同步开源了 FengShen(封神) 框架。我们参考了 HuggingFace, Megatron-LM, Pytorch-Lightning, DeepSpeed 等优秀的开源框架，结合 NLP 领域的特点, 以 Pytorch 为基础框架，Pytorch-Lightning 为 Pipeline 重新设计了 FengShen。FengShen 可以应用在基于海量数据 (TB 级别数据) 的大模型 (百亿级别参数) 预训练以及各种下游任务的微调，用户可以通过配置的方式很方便地进行分布式训练和节省显存的技术，更加聚焦在模型实现和创新。同时 FengShen 也能直接使用 HuggingFace 中的模型结构进行继续训练，方便用户进行领域模型迁移。FengShen 针对封神榜开源的模型和模型的应用，提供丰富、真实的源代码和示例。随着封神榜模型的训练和应用，我们也会不断优化 FengShen 框架，敬请期待。