提交指南

文件结构

参赛者需要将每个任务的预测文件以及模型定义文件分开存放，一个任务一个文件夹，总共五个文件夹，每个文件夹需以任务名称命名，然后将这个五个文件夹打包在一个 zip 压缩文件中，具体的文件夹结构如下：

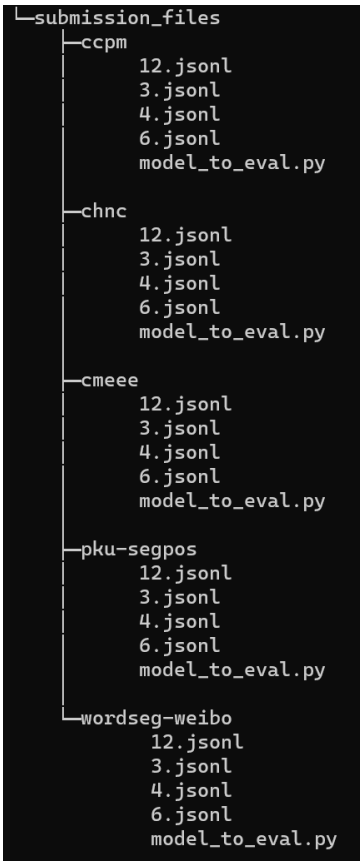

图 1：文件夹结构

注意：预测文件必须为 jsonl 格式，模型定义文件必须为 python 文件，每个任务可以有多个预测文件（每个预测文件对应不同 FLOPs 下的预测结果，此种情况一般发生在动态方法上，如 Early exiting）。

预测文件格式

每个任务的预测文件格式如下示例所示：


```
{"answer": 0, "modules": "(88),emb;(88,768),layer_1;(88,768),layer_2;(88,768),layer_3;(88,768),pooler;(768),classifier"}
{"answer": 0, "modules": "(84),emb;(84,768),layer_1;(84,768),layer_2;(84,768),layer_3;(84,768),pooler;(768),classifier"}
{"answer": 2, "modules": "(84),emb;(84,768),layer_1;(84,768),layer_2;(84,768),layer_3;(84,768),pooler;(768),classifier"}
```


图 2：CCPM 任务预测文件示例


```
{"label": "finance", "modules": "(21),emb;(21,768),layer_1;(21,768),layer_2;(21,768),layer_3;(21,768),pooler;(768),classifier"}
{"label": "food", "modules": "(15),emb;(15,768),layer_1;(15,768),layer_2;(15,768),layer_3;(15,768),pooler;(768),classifier"}
{"label": "society", "modules": "(24),emb;(24,768),layer_1;(24,768),layer_2;(24,768),layer_3;(24,768),pooler;(768),classifier"}
```


图 3：CHNC 任务预测文件示例


```
{"entities": [{"start_idx": 13, "end_idx": 15, "type": "equ", "entity": "质谱仪"}], "modules": "(57),emb;(57,768),layer_1;(57,768),layer_2;(57,768),layer_3;(57,768),classifier"}
{"entities": [{"start_idx": 82, "end_idx": 84, "type": "bod", "entity": "碰撞室"}], "modules": "(143),emb;(143,768),layer_1;(143,768),layer_2;(143,768),layer_3;(143,768),classifier"}
{"entities": [{"start_idx": 61, "end_idx": 65, "type": "dis", "entity": "遗传代谢病"}], "modules": "(86),emb;(86,768),layer_1;(86,768),layer_2;(86,768),layer_3;(86,768),classifier"}
```


图 4：CMeEE 任务预测文件示例


```
{"span_list": [{"start": 0, "end": 1, "type": "ad"}], "modules": "(8),emb;(8,768),layer_1;(8,768),layer_2;(8,768),layer_3;(8,768),classifier"}
{"span_list": [{"start": 0, "end": 2, "type": "w"}], "modules": "(17),emb;(17,768),layer_1;(17,768),layer_2;(17,768),layer_3;(17,768),classifier"}
{"span_list": [{"start": 0, "end": 1, "type": "r"}], "modules": "(5),emb;(5,768),layer_1;(5,768),layer_2;(5,768),layer_3;(5,768),classifier"}
```


图 5：PKU-SEGPOS 任务预测文件示例


```
{"ans": "你 拥有 这么 多 ， 凭 什么 说 你 一无所有 呢 ？ ”", "modules": "(19),emb;(19,768),layer_1;(19,768),layer_2;(19,768),layer_3;(19,768),classifier"}
{"ans": "你 是否 曾经 炒 过 某 一样 ？", "modules": "(11),emb;(11,768),layer_1;(11,768),layer_2;(11,768),layer_3;(11,768),classifier"}
{"ans": "来 一 波 好 礼 会 倍感 精神 ！", "modules": "(11),emb;(11,768),layer_1;(11,768),layer_2;(11,768),layer_3;(11,768),classifier"}
```


图 6：WordSeg-Weibo 任务预测文件示例

每个样例都必须包含两项：预测的结果和预测该样例所经过的模型中的模块（对应示例中的“modules”）。所有的模块都必须在所提交的模型文件中有单独的定义。

模型文件

每个任务的文件夹下都必须包含一个用于定义模型的 python 文件，在预测文件中出现的 module 都必须在该文件中有单独的定义，下图是一个示例：

```
# import packages
import torch.nn as nn
from transformers import BertConfig
...

# module definitions
class ElasticBERTEmbeddings(nn.Module):
    def __init__():
        ...
    def forward(x):
        ...

class ElasticBERTLayer(nn.Module):
    def __init__():
        ...
    def forward(x)
        ...

class ElasticBERT(nn.Module):
    def __init__():
        ...
    def forward(x)
        ...

# module dict
config = BertConfig(num_labels=2)
module_list = {
    'emb': ElasticBertEmbeddings(config),
    'layer_1': ElasticBertLayer(config),
    'exit_1': nn.Linear(config.hidden_size, num_labels),
    'layer_2': ElasticBertLayer(config),
    'exit_2': nn.Linear(config.hidden_size, num_labels),
    ...
}
entire_model = ElasticBERT(config)
```

图 7：模型文件示例

注意：尽量在定义时将不同的模块分离开，以保证每一个模块都可以单独运行成功；此外，模块字典必须命名为 ”module_list”，且需要将所有模块都添加进模块字典中；每个模块的名字尽量不要命名过长，以免增加预测文件的大小。

运行环境

由于在评测过程中需要运行模型，所以对于运行环境有以下要求：

- 定义模型必须使用 Pytorch 深度学习库，且 Pytorch>=1.8.1
- 若使用 transformers 库，则 transformers>=4.6.1