本周进度

- **Demo && 界面**
- **面向对象方法学课程大作业**

Demo

- **模型使用**
  - **BertForMaskedLM**
  - **bert-base-chinese**
- **基本思路**
  1. 输入句子，将待预测的文字使用`[MASK]`进行标识;
  2. 对句子进行`encode`，得到对应 tensor;
  3. 使用`bert-base-chinese`模型进行预测， 得到结果`prediction_scores`;
  4. 对`prediction_scores`做`softmax`;
  5. 对 softmax 结果取前五个最大值，进行`decode`,得到预测词;
- **交互界面**

- **核心代码**
  - `config 配置`

```
import os
from transformers import BertTokenizer, BertModel,
BertForMaskedLM, BertConfig


ROOT = os.environ['HOME']  # 用户目录

MODEL_DIR = os.path.join(ROOT, 'Sources/bert-base-chinese')  # 模
型目录

CONFIG = BertConfig.from_pretrained(MODEL_DIR +
'/bert_config.json')  # 加载config

TOKENIZER = BertTokenizer.from_pretrained(MODEL_DIR)  # 加载
tokenizer

MODEL = BertForMaskedLM.from_pretrained(MODEL_DIR, config=CONFIG)
 # 加载模型
```

- 预测代码

```
class PerdictWord(models.Model):

    @staticmethod
    def get_words(sentence):
        # 加载模型
        tokenizer = config.TOKENIZER
        model = config.MODEL

        sentence = sentence.replace('[MASK]', '&')

        input_ids =
torch.tensor(tokenizer.encode(sentence)).unsqueeze(0)

        # 对句子中的[MASK]标签进行替换
        index = sentence.find('&')
        input_ids[0][index] = tokenizer.mask_token_id

        # 切换到 gpu 上运行
        if torch.cuda.is_available():
            input_ids = input_ids.to('cuda')
            model.to('cuda')

        outputs = model(input_ids, masked_lm_labels=input_ids)
        loss, prediction_scores = outputs[:2]

        # 对预测后的分数做 softmax 取前5个最大值
        sm_result = F.softmax(prediction_scores, dim=2)
        topk_values, topk_indices = sm_result.topk(5, dim=2)[:2]
```

```
            # 取出预测词 values 和 indices
            topk_values = topk_values[0][index]
            topk_indices = topk_indices[0][index].tolist()  # 预测词转
    换为列表，用于.decode

            perdict_words = []
            for i, indice in enumerate(topk_indices):
                # 输出预测词和概率
                word = {'word': tokenizer.decode(
                    indice), 'percent': topk_values[i].item()}
                perdict_words.append(word)
            return perdict_words
```

下周任务

- 论文阅读
- 课程作业 && 复习
- 课程学习