

本周进度

- Demo优化修改
- 面向对象方法学课程大作业
- 考试复习

Demo

- 添加多个 [MASK] 标签预测代码，发现代码逻辑错误并修改
- 错误总结
 - [MASK]标签定位错误
 - 使用 bert-base-chinese 进行 encode 在会过滤掉空格并在句子前后添加 [CLS]、[SEP] 标签，之前直接获取 [MASK] 在输入句子中的 index 值，最终输出，实际和正确的下标相差 1 位

```
[02/000/2020-11-17-18] 021 / 111 / 212 200 1020
输入句子：你 [MASK]，我 [MASK]小 [MASK]。
encode后句子：[CLS] 你 [MASK] ， 我 [MASK] 小 [MASK] 。 [SEP]
```

- 交互界面

BertForMaskedLM 中文预测

— 使用 BertForMaskedLM 模型对输入的句子 [MASK] 标签进行预测

你[MASK]，我[MASK]小[MASK]。

提交

预测结果：

好 84.90%	是 37.17%	孩 1.56%
说 3.05%	叫 33.70%	明 1.35%
看 2.55%	的 3.98%	宝 1.35%
是 0.74%	爱 3.78%	强 1.25%
们 0.63%	和 1.53%	白 1.20%

- 核心代码

- 预测代码

```
class Prediction(models.Model):

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

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

        index_list = []
```

```

for i, id in enumerate(input_ids[0]):
    if id == 103:
        index_list.append(i)

# 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
mask_values = (topk_values[0][index_list]).tolist()
mask_indices = (topk_indices[0][index_list]).tolist()

# 将预测词 decode
for i, indices in enumerate(mask_indices):
    mask_indices[i] = tokenizer.decode(indices).split(' ')

return {'values': mask_values, 'indices': mask_indices},
len(index_list)

```

下周任务

- 论文阅读
- 使用 **BERT** 其他模块编写Demo
- 课程学习