

深度学习

Lab10-VisualCGEC task

陈沁文 王千予

这是期末大作业，请在6月9号结束之前提交！

背景

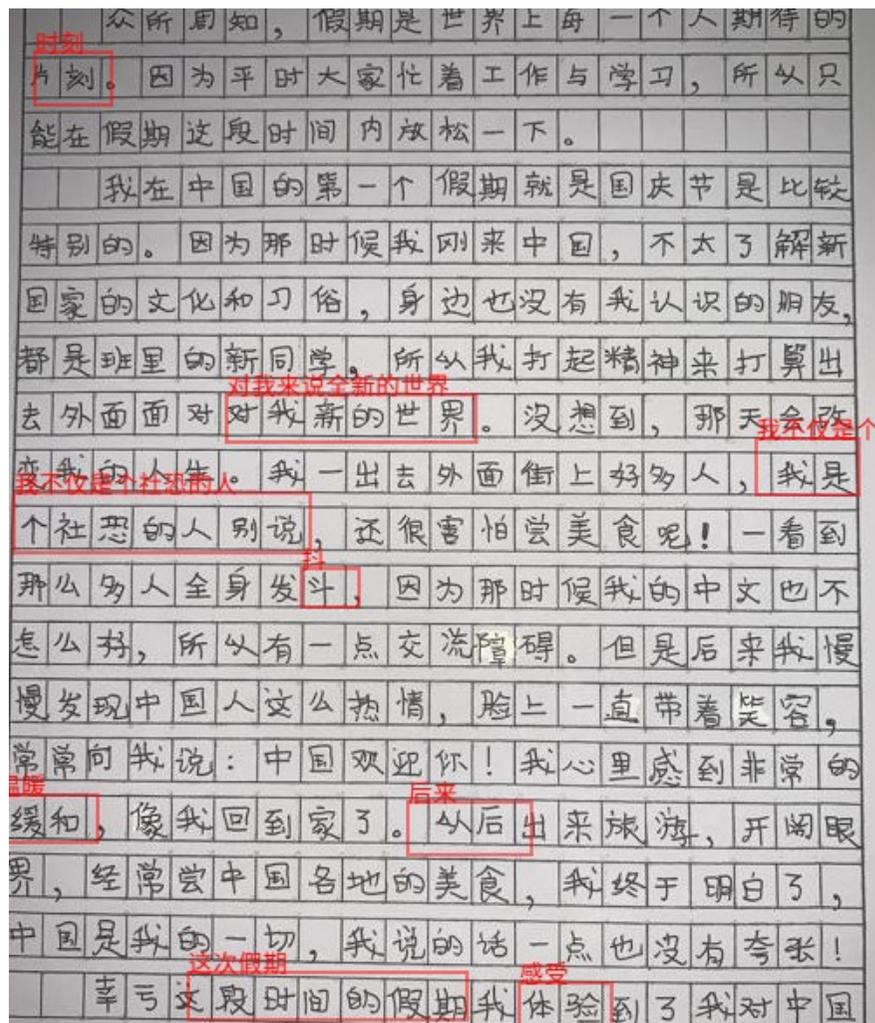
- 视觉汉语语法纠错（Visual Chinese Grammatical Error Correction, VisualCGEC）是一项结合图像与文本信息，自动检测并纠正中文文本中语法错误的任务
- 与传统的中文语法纠错任务不同，VisualCGEC 引入了图像信息（如 OCR 识别结果、文本在图像中的位置等），识别并纠正中文文本中的语法错误，生成语法正确、语义通顺的文本输出

input_text	predict_text
文本纠错：\n少先队员因该为老人让坐。	少先队员应该为老人让座。

文本纠错示例

VisualCGEC task

- 任务一
 - 给定中文手写作文图片，输出纠正后的文本
- 任务二
 - 返回语法纠错对应图片上的所有边界框 (bounding box) 位置



上图红框即bounding box

Dataset

数据集	介绍	条数
train_data.json	训练集, 给定图片id、图片路径、源文本、目标文本和边界框列表	350
test_data.json	测试集, 给定图片id和图片路径	88

train_data.json文件结构 :

```
[
  {
    "fk_homework_id": xxx, // 图片id
    "path": "xxx.xxx", // 图片路径
    "source_text": "xxx", // ocr识别文本结果 (源文本)
    "target_text": "xxx", // gec目标输出 (目标文本)
    "bounding_box_list": // 边界框列表
      [
        {"start_x": xxx, // start_x表示边界框最小横坐标
         "end_x": xxx, // end_x表示边界框最大横坐标
         "start_y": xxx, // start_y表示边界框最小纵坐标
         "end_y": xxx}, // end_y表示边界框最大纵坐标
        ...]
      ],
    ...
  },
  ...
]
```

学生需提交的predict.json文件结构 :

```
[
  {
    "fk_homework_id": xxx,
    "path": "xxx.xxx",
    "source_text": "xxx",
    "predict_text": "xxx", // gec预测输出
    "bounding_box_list": // 预测的边界框列表,
                        // 不做任务二则设置为空列表[]
      [
        {"start_x": xxx,
         "end_x": xxx,
         "start_y": xxx,
         "end_y": xxx},
        ...]
      ],
    ...
  },
  ...
]
```

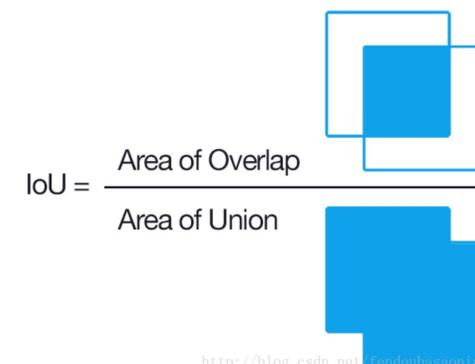
注意：在目标检测中，坐标体系的零点坐标通常为图片的左上角，X轴为图像矩形上面的水平线；Y轴为图像矩形左边的垂直线。

Evaluation

- 评估指标：0.5*F0.5 + 0.5*IoU
 - F-score:计算文本和源文本之间准确率和召回率的一个综合指标
 - 当 $\beta=1$ 时，即F1，准确率和召回率一样重要
 - 当 $\beta=0.5$ 时，即F0.5，更重视准确率

$$F_{\beta} = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

- IoU (Intersection over Union)
 - IoU是一种测量在特定数据集中检测相应物体准确度的一个标准，只要是在输出中得出一个预测范围的任务都可以用IoU来进行测量
 - IoU=预测bounding box和目标bounding box的交集面积/并集面积
 - 在本次任务中，预测bounding box对应图片中预测需要纠错的源文本



注意：如果不做任务二则IoU分数为0，该评估指标不代表期末作业分数，仅作为榜单分数

Evaluation代码

```
# F0.5
def compute_f05_char_level(ref, pred):
    ref_chars = set(ref)
    pred_chars = set(pred)
    correct = len(ref_chars & pred_chars)
    pred_total = len(pred_chars)
    ref_total = len(ref_chars)
    if pred_total == 0 or ref_total == 0:
        return 0.0
    precision = correct / pred_total
    recall = correct / ref_total
    beta = 0.5
    return (1 + beta**2) * precision * recall / (beta**2 * precision + recall) if
    (precision + recall) > 0 else 0.0
```

```
# IOU计算
def compute_iou(box1, box2):
    x_left = max(box1["start_x"], box2["start_x"])
    y_top = max(box1["start_y"], box2["start_y"])
    x_right = min(box1["end_x"], box2["end_x"])
    y_bottom = min(box1["end_y"], box2["end_y"])
    #如果没有重叠
    if x_right <= x_left or y_bottom <= y_top:
        return 0.0
    inter_area = (x_right - x_left) * (y_bottom - y_top)
    area1 = (box1["end_x"] - box1["start_x"]) * (box1["end_y"] - box1["start_y"])
    area2 = (box2["end_x"] - box2["start_x"]) * (box2["end_y"] - box2["start_y"])
    union_area = area1 + area2 - inter_area
    return inter_area / union_area if union_area > 0 else 0.0
```

```
def compute_acc():
    with open(truth_file, 'r', encoding='utf-8') as f:
        test_data = json.load(f)
    with open(submission_answer_file, 'r', encoding='utf-8') as f:
        pred_data = json.load(f)

    pred_map = {item["fk_homework_id"]: item for item in pred_data}

    total_score = 0.0
    total_count = len(test_data)

    for gt in test_data:
        fkid = gt["fk_homework_id"]
        if fkid not in pred_map:
            f05 = 0.0
            iou_score = 0.0
        else:
            pred = pred_map[fkid]
            f05 = compute_f05_char_level(gt["target_text"], pred["predict_text"])

            gt_boxes = gt.get("bounding_box_list", [])
            pred_boxes = pred.get("bounding_box_list", [])
            iou_score = 0.0
            if pred_boxes:
                ious = []
                for pb in pred_boxes:
                    #一对一计算IOU
                    max_iou = max(compute_iou(pb, gb) for gb in gt_boxes)
                    ious.append(max_iou)
                iou_score = sum(ious) / len(ious) if ious else 0.0

            #加权求和
            final_score = 0.5 * f05 + 0.5 * iou_score
            total_score += final_score

    acc = total_score / total_count if total_count > 0 else 0.0

    # 输出写入文件 (不要改)
    with open(output_filename, 'w') as output_file:
        output_file.write("ACC: %0.4f\n" % acc)

    print('The ACC on test data is %f' % acc)
```

Baseline

- OCR：调用paddleocr识别图片文字
 - PaddleOCR是百度开源的超轻量级文字识别模型/工具库，提供了数十种文本检测、识别模型，用户可以自定义训练
- GEC模型：由people2014corpus_chars.klm模型直接推理
 - 140M，由2014版人民日报数据训练的模型，已放在天梯作业目录/models下，来源<https://huggingface.co/shibing624/chinese-kenlm-klm>
- 具体代码在天梯上的baseline.ipynb，可直接运行，运行Baseline后提交的结果：

#	SCORE	FILE NAME	SUBMIT TIME	FILE SIZE(KB)	STATUS	✓	
1	0.2635	prediction.zip	2025/04/24 08:07:39	62742	Finished	✓	+

可以考虑的改进

- 尝试图片数据预处理，提高识别效果
- 可尝试现成的ocr模型/接口，比较识别效果
- ocr/gec模型可在预训练模型基础上基于所给训练集进行微调
- ...

不限于上述，只是提供参考，大家可以自行选择高效的方法，提高ocr和gec的效果

注意事项

- 可选择适合自己的平台跑实验，最终预测文件predict.json需压缩成prediction.zip，并**存放**
在天梯作业的output/下，再在jupyter上点击submit按钮完成提交
 - 若代码是在天梯上运行，压缩操作的实现可参考baseline.ipynb最后
 - 实现的代码需要上传到天梯，代码一定要是完整的、可复现的
- 天梯**每日提交次数限制为5次**
- 完成一篇实验报告，上传到天梯作业根目录下，按“实验报告_姓名_学号.pdf”的格式命名
 - **最多2页的pdf，不用贴代码，仅说明核心方法**
- 可随意选择python依赖包、模型进行模型搭建和训练
- 不可以利用额外的数据集，不可以手动标注测试集

评分标准

榜单分值会作为主要打分依据，但还会考虑代码的规范程度、方案创新程度、代码工作量对分数(总分100)进行-10~+10分的调整。