2025 自然语言处理
课程设计 1

人工智能学院 221300079 王俊童

2025.4.1

综述，首先观察代码结构，逻辑如下：

- 命令行参数解析。有 method，是否 analyze，statistical 里面方法的选取。
- 加载数据和数据分析（需要我们实现数据分析）
- 三个方法的训练：
  - rule: 基于一些规则得到的一个实现。train 有四种纠错规则：
    - $\_extract\_confusion\_pairs$：字符混淆对提取。
    - $\_extract\_punctuation\_rules$：标点符号规则提取
    - $\_extract\_grammar\_rules$：语法规则提取
    - $\_extract\_word\_confusion$：词汇混淆对提取

    然后以上四种错误的纠错发生在 correct 里面。
  - statistical：基于统计学习方法的纠错。这个里面又分为两个模型:
    - ngram 模型：初始化了一堆数据结构，1-4 的 gram 方法，字符混淆矩阵和错误率等
    - ml 模型：用机器学习方法去做。
  - 集成学习方法，在框架代码的 ensemble 部分有留给我们实现。
- 三个方法对应的纠错和评估。跟上面一样了，可以实现很多的 correct 方法，都有对应接口。

可以看出整个代码框架都还是比较整齐的，我们需要完成的 TODO 任务如下:

- 数据的 analyze 分析部分和画图。
- rule：完成规则方法的实现。完成对应规则方法的纠错改正。
- statistical：完成 ngram 和 ml 方法的对应修正和改正。
- main：完成集成学习方法。
- 其余可以加一些深度学习之类的方法实现。

1 实现方法及其简单描述，遇到的问题和解决方案（全包含，就不单独列了，按照我的编程和问题思考思路来写的）

1.1 数据分析部分

数据分析部分，我们将原来的 args 做了一点点修改，然后我们首先可以根据原词典数据进行统计，把 label 为 1 的错误数据中的错误字符全部统计出来，而且可以得到错误率最高的 10 个的错误模式和错误字符，这更方便我们后续处理：

```
# 只看错的
if label == 1:
    error_count += 1

    if len(source) == len(target):
        for i, (s_char, t_char) in enumerate(zip(source, target)):
            if s_char != t_char:
                char_error_freq[s_char] += 1
                error_patterns[(s_char, t_char)] += 1
```

然后我们把它可视化, 同时，由于 matplotlib 不支持中文字体，需要更换自己电脑里面的路径。这个在对应 data analysis 的 python 文件里面有讲。

可以看一个我做出来的效果，还是蛮不错的。可以看到的和地的错误最多，还有的和得之类的，一般都

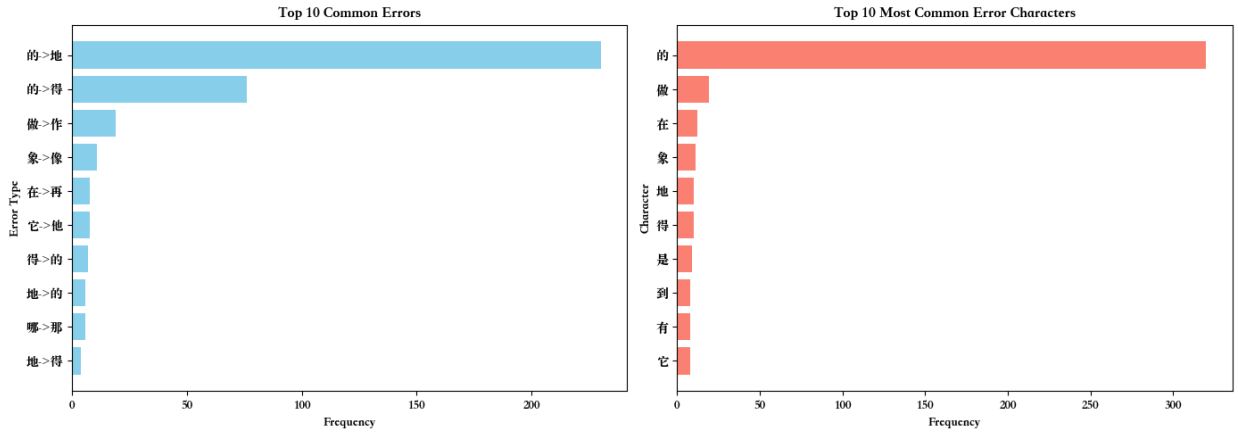

Figure 1: error distribution

是些同音不同意的字。

1.2 3 个方法部分

1.2.1 方法 1:rule

rule 这个方法还蛮简单的，基本是基于人类的常识性的方法，有点像是打表。但是肯定有补全不了的规则，这个是硬伤。共有如下的需要填补的方法:

- self._extract_confusion_pairs：这个方法已经给我们补全了。意思是提取了混淆字符对。但是这个一眼就存在一些问题:
  - 没有考虑插入和删除的错误
  - 没有考虑很强烈的上下文特征
  - 不同的 count 对于噪声过滤效果不一样，可以产生不一样的效果

  我们首先修改这个混淆对的做法:
- self._extract_punctuation_rules
- self._extract_grammar_rules
- self._extract_word_confusion

1.3 其余方法

2 如何复现结果和代码环境依赖问题

3 不同实验方法的对比结果

4 一些简单思考