IRBL迭代一：软件详细设计

特别声明：内部资料，请勿传播

作者：程荣鑫

更新历史：

更新日期	更新原因	责任人
2021.3.13	创建文档	程荣鑫
2021.3.19	更新ResultPrinter模块的接口设计	程荣鑫

1. 引言

1.1 编写目的和范围

编写目的：完善前作《IRBL迭代一：软件概要设计》的软件设计细节，将设计落实到编码层面，发挥设计文档的指导作用。

范围：本文档包含IRBL项目迭代一中的全局数据结构、总体设计、模块设计、接口设计和数据传输设计。受众为软工三团队KhyYYDS小组的全体成员。

1.2 术语表

术语	含义
IRBL	基于信息检索的缺陷定位系统

1.3 参考资料

《IRBL迭代一：软件概要设计》、《IRBL迭代一：项目启动》以及moodle上的IRBL基于信息检索的错误定位材料集

1.4 使用的文字处理和绘图工具

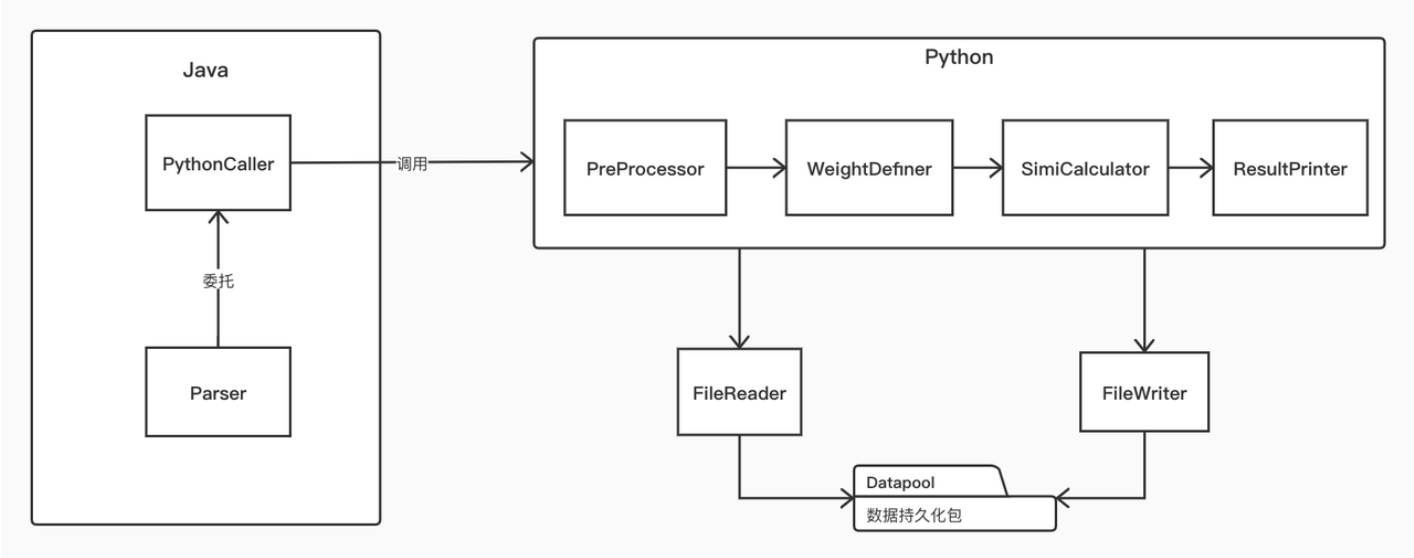
绘图：ProcessOn

文字处理：飞书云文档

2. 全局数据结构说明

数据结构	类型	含义	持久化格式
word2idx	dict<str, int>	对词语集进行编号的结果	json
doc2vec	dict<str, list>	每篇文本对应的词向量	json
simi_matrix	dict<str, dict<str, float>>	bug报告和代码文件的相似度	json

3. 总体设计



总体设计相比概要设计文档中的设计内容大体没有发生变化，仍然是流水线式设计，只是将原本图中与File System中的交互细化为，FileReader与FileWriter模块和Datapool文件夹的交互，Python模块统一通过FileReader与FileWriter存取Datapool中的数据文件。

4. 模块设计

考虑的实现的便利性，模块设计和《IRBL迭代一：软件概要设计》中的模块设计稍有出入，如有冲突，请以**本文档**的的设计为准。

- Parser

职责：解析命令，委托PythonCaller执行相应程序

支持命令（同时支持简写命令）：

preprocess(简写：p, /p): 调用code_preprocessor.py和bug_preprocessor.py，执行文本预处理任务

defineWeight/define weight(简写：dw, /d): 调用weight_definer.py，执行权重计算和文本向量化的任务

calculateSimilarity(简写：cs, /c): 调用simi_calculator.py，执行相似度计算任务

printResult(简写：pr, /pr): 调用result_printer.py，执行运行结果打印的任务

doall(简写：all, /a): 依次调用code_preprocessor.py，bug_preprocessor.py, weight_definer.py, simi_calculator.py, result_printer.py，即完整地走完流水线

exit(简写：q): 退出IRBL主程序

- PythonCaller

  职责：根据Parser的要求调用相应的python程序

- PreProcessor

  职责：执行预处理任务

  包含code_preprocessor.py和bug_preprocessor.py

- WeightDefiner

  职责：执行词语权重计算任务，并将文档向量化

  由weight_definer.py实现

- SimiCalculator

  职责：执行相似度计算任务，并把结果保存下来

  由simi_calculator.py实现

- ResultPrinter

  职责：输出相似度计算的结果及相关指标

  由result_printer.py实现

- FileReader

  职责：读取Datapool中文件的内容

  由file_reader.py实现，其中包含基类FileReader，派生类JSONReader和NpyReader

- FileWriter

职责：向Datapool中文件写入内容

由file_writer.py实现，其中包含基类FileWriter，派生类JSONWriter和NpyWriter

5. 接口设计

5.1 Parser

供接口：无

需接口：PythonCaller(String pyPath, String[] args), PythonCaller.exec()

5.2 PythonCaller

供接口：

1. PythonCaller(String pyPath, String[] args)
2. PythonCaller.exec()

需接口：

1. code_processor.main
2. bug_processor.main
3. weight_definer.main
4. simi_calculator.main
5. result_printer.main

5.3 PreProcessor

供接口：

1. code_processor.main
2. bug_processor.main

需接口：

1. FileWriter.writeFile

   PreProcessor模块需要使用FileWriter写入txt文件

5.4 WeightDefiner

供接口：

1. weight_definer.main

需接口：

1. FileReader.readFile()
   WeightDefiner需要用到FileReader读取.txt文件

5.5 SimiCalculator

供接口：

1. simi_calculator.main

需接口：

1. FileReader.readFile()
   SimiCalculator需要用到JSONReader
2. FileWriter.writeFile()
   SimiCalculator需要用到JSONWriter

5.6 ResultPrinter

供接口：

1. result_printer.main
   包含以下子流程：
   get_top_K:
   获取与每个BUG报告前K个最相关的代码文件名（不含.java后缀）
   print_top_K:
   输出与每个BUG报告前K个最相关的代码文件名（不含.java后缀）
   print_metrics:
   输出top1、top5、top10、MRR、MAP指标

需接口：

1. FileReader.readFile()
   ResultPrinter需要用到XlsReader和JSONReader

5.7 FileReader

供接口：

1. FileReader.readFile()：读取文件，返回文件内容

   由FileReader的子类提供不同的实现，JSONReader返回dict，NpyReader返回ndarry，FileReader本身有默认实现（返回str）

需接口：无

5.8 FileWriter

供接口：

1. FileWriter.writeFile()：向文件写内容

   由FileWriter的子类提供不同的实现，JSONWriter写入dict，存为json文件；NpyWriter写入ndarry，存为.npy文件

需接口：无

6. 数据传输设计

数据传输主要发生在python包中，考虑到模块间数据传输的量非常大，我们采用文件读写的方式存取数据。

1. 所有的数据文件都在src/main/python/irbl/Datapool文件夹中
2. python程序统一通过file_reader.py和file_writer中的接口读取数据