医学知识图谱创建工具

团队项目总结

杨璇 李泰郎

2017

团队项目总结

目录

[一. 敏捷冲刺每日报告 3](#_Toc497761475)

[1. 敏捷冲刺安排 3](#_Toc497761476)

[2. 敏捷冲刺每日报告 3](#_Toc497761477)

[1) 模块以及领取人 3](#_Toc497761478)

[2) 敏捷冲刺每日报告 3](#_Toc497761479)

[3) 燃尽图 4](#_Toc497761480)

[二、测试报告 4](#_Toc497761481)

[1. 测试过程 4](#_Toc497761482)

[2. 发现的bug 6](#_Toc497761483)

[3. 场景测试 7](#_Toc497761484)

[4. 测试矩阵 7](#_Toc497761485)

[5. 出口条件 7](#_Toc497761486)

[1) 系统运行环境 7](#_Toc497761487)

[2) 系统界面显示 8](#_Toc497761488)

[3) 系统功能需求 8](#_Toc497761489)

[三、alpha版本发布说明 9](#_Toc497761490)

[1. 本版本的功能 9](#_Toc497761491)

[2. 功能的图片、视频展示 9](#_Toc497761492)

[3. 运行环境要求 9](#_Toc497761493)

[4. 安装方法 9](#_Toc497761494)

[5. 系统已知问题和限制 10](#_Toc497761495)

[6. 发布（并未实际发布） 10](#_Toc497761496)

[1) 发布方式 10](#_Toc497761497)

[2) 发布地址 10](#_Toc497761498)

[四、小结 10](#_Toc497761499)

[1. 源码仓库地址 10](#_Toc497761500)

[2. 过程回顾 10](#_Toc497761501)

[1) 项目预期 10](#_Toc497761502)

[2) 满足的需求 11](#_Toc497761503)

[3) 一次迭代完成的目标 12](#_Toc497761504)

[4) 团队分工协作的经验及教训 12](#_Toc497761505)

[5) 项目管理的方法 12](#_Toc497761506)

[6) 时间/质量/资源的平衡 12](#_Toc497761507)

[3. 团队项目的实际进展 13](#_Toc497761508)

[4. 团队成员的角色和具体贡献 13](#_Toc497761509)

[5. 下一阶段的展望 14](#_Toc497761510)

## 一. 敏捷冲刺每日报告

1. 敏捷冲刺安排

在10月27日至11月6日的时间内，安排从10月27日至10月31日连续5天的敏捷冲刺。

1. 敏捷冲刺每日报告

###### 模块以及领取人

|  |  |
| --- | --- |
| 模块 | 领取人 |
| #1界面的设计 | 陈健 |
| #2界面的开发 | 陈健 |
| #3表单提交逻辑 | 蒋文佳 |
| #4后台数据接收逻辑 | 陈健 |
| #5绘图逻辑 | 陈健 |
| #6接收文章并存储 | 杨璇 |
| #7获取算法结果,返回前端 | 杨璇 |
| #8设计正则表达式 | 乔雨 |
| #9处理文章，存储为文章预处理结果对象 | 乔雨 |
| #10构建本体库 | 李泰郎 |
| #11自然语言处理。生成文章处理结果对象 | 李泰郎 |
| #12将文章处理结果对象以json格式存储返回 | 李泰郎 |
| #13模块6、7与前端整合 | 李泰郎、陈健 |
| #14模块6、7与模块8、9整合 | 李泰郎 |
| #15所有模块的整合 | 杨璇 |

###### 敏捷冲刺每日报告

<敏捷冲刺每日报告.xlsx>

###### 燃尽图

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 10月27日 | 10月28日 | 10月29日 | 10月30日 | 10月31日 | 11月1日 |
| 剩余时间 | 41 | 24.2 | 17.7 | 7.8 | 7 | 0 |
| 实际花费时间 | 15.8 | 22.3 | 32.2 | 35.2 | 38.2 | 39.2 |
| 预估剩余时间 | 41 | 32 | 24 | 16 | 8 | 0 |

## 二、测试报告

1. 测试过程

用例1

|  |  |
| --- | --- |
| 模块 | 前端用户界面模块 |
| 实现功能 | 复制文章内容进入输入框 |
| 输入 | 文章标题，文章内容 |
| 实际操作 | 用户将文章的标题和内容提交到网页中对应的文本框中 |
| 期待结果 | 能够正确的显示用户输入的文章标题与内容 |

用例2

|  |  |
| --- | --- |
| 模块 | 前端用户界面模块 |
| 实现功能 | 清除用户输入的文章内容 |
| 输入 | 文章标题，文章内容 |
| 实际操作 | 1. 用户将文章的标题和内容提交到网页中对应的文本框中 2. 点击“重置”按钮 |
| 期待结果 | 清空所有的输入内容 |

用例3

|  |  |
| --- | --- |
| 模块 | 前端用户界面模块 |
| 实现功能 | 确保用户提交到后端的内容不为空 |
| 输入 | 文章标题、文章内容之一或两者都没有 |
| 实际操作 | 1. 用户只输入文章的标题或者内容，或者两个内容都不输入 2. 点击“提交”按钮 |
| 期待结果 | 显示提示内容“**文章标题或内容不能为空！**”，需要完善输入 |

用例4

|  |  |
| --- | --- |
| 模块 | 前端用户界面模块 |
| 实现功能 | 后台服务器出bug，前端提示提交失败 |
| 输入 | 文章标题、文章内容 |
| 实际操作 | 1. 用户将文章的标题和内容提交到网页中对应的文本框中 2. 点击“提交”按钮 3. 此时后台计算失败或者后台服务器没有启动 |
| 期待结果 | 显示提示内容“**提交失败，请稍后重试！**” |

用例5

|  |  |
| --- | --- |
| 模块 | 后端正则表达式模块 |
| 实现功能 | 对于输入的一篇文章，通过构造的正则表达式，将文章分解成不同等级标题的内容 |
| 输入 | 一个存储了文章的txt文件 |
| 实际操作 | 1. 将文章存储到txt文件中 2. 调用正则表达式模块 3. 获取到不同标题的对象的List，这个List按照文章的标题顺序存储 |
| 期待结果 | 获取到不同标题的对象的List |

用例6

|  |  |
| --- | --- |
| 模块 | 后端THULAC词法分词模块 |
| 实现功能 | 对于一个字符串的内容，通过词法分析模块分析成字符串类型的List |
| 输入 | 一个字符串 |
| 实际操作 | 1. 将字符串存储到临时文本文件中 2. 调用词法分析模块读取该文本文件 3. 获取到分解之后分词的字符串的List |
| 期待结果 | 分解之后分词的字符串的List |

用例7

|  |  |
| --- | --- |
| 模块 | 后端本体数据库访问模块 |
| 实现功能 | 对于分解之后分词的字符串的List，去对应标题下的数据库中查找是否存在对应的本体 |
| 输入 | 分解之后分词的字符串的List |
| 实际操作 | 1. 获取到分词之后分词的字符串的List 2. 将这个List中组合成二个一组、三个一组、四个一组的List 3. 去数据库中查找这个字符串List对应的标题下存在的本体，返回成一个List 4. 将这四个List中的内容，去数据库返回的List中查找，查找是否存在对应的本体 5. 将结果保存成字符串的List |
| 期待结果 | 本体字符串的List |

用例8

|  |  |
| --- | --- |
| 模块 | 后端本体数据库访问模块 |
| 实现功能 | 将数据库提取出来的本体字符串List的信息包装成新的下一级标题对象，将该对象插入到原来的标题对象List中 |
| 输入 | 本体字符串的List |
| 实际操作 | 1. 数据库返回的本体字符串的List 2. 包装成新的下一级标题对象 3. 插入到原来的标题对象List中本体字符串对应的上一级标题中对应的位置 |
| 期待结果 | 成功插入到原有的标题对象List中 |

用例9

|  |  |
| --- | --- |
| 模块 | 后端本体构建模块 |
| 实现功能 | 将标题对象List按照树的结构存储成json格式的内容 |
| 输入 | 标题对象List中 |
| 实际操作 | 1. 将标题对应的List按顺序读取，存储成树状结构 2. 将该树状结构存储成json格式的内容 |
| 期待结果 | json格式的结果 |

用例10

|  |  |
| --- | --- |
| 模块 | 后端服务器接口模块 |
| 实现功能 | 获取到前端的页面提交的内容，按代码逻辑调用后台各个端口获取到结果返回到前端 |
| 输入 | 前端的请求 |
| 实际操作 | 1. 获取到前端请求 2. 返回json结果 |
| 期待结果 | 正确的执行获取与返回 |

1. 发现的bug

|  |  |  |
| --- | --- | --- |
| a.后端返回json无法正确解析 | | |
| b.后端tomcat服务器异常，无法找到Servlet地址 | | |
| c.前端显示结果图连线过多 | | |
| d.正则表达式模块无法正确分段 | | |
| e.判断四级标题（1.疾病治疗）文本的正则表达式错误，将正文中含有1.X形式的文本识别为标题 | | |
| f.判断四级标题（1.疾病治疗）文本的正则表达式错误，免识别正文，但同时也识别不了标题 | | |
| g.四级标题在VSCODE测试中可以匹配成功，换成JAVA正则仍旧失败 | | |
| h.结构化存储多个普通文本不能存入所属级别的节点中，每次都初始化Node就会出现普通文本也存为一个节点的问题 | | |
| i.用于存储用户提交文章的临时文件找不到文件的路径 | | |
| j.使用THULAC分词工具进行分词的时候找不到存放models文件的文件夹，无法进行分词 | | |
| k.THULAC分词工具进行分词的时候有些对应的内容无法分词 | | |
| l.读取本体数据库之后插入文章标题对象时无法正确插入到正确位置 | | |
| m将分词最终结果组合成多个词语的结合词的顺序错误 | | |
| n程序存在内存泄漏问题，影响到程序的持续运行 | | |
| o.程序词法分析模块以及数据库访问模块运行速度较慢，影响整体项目运行速度 | | |
| p.前端获取到后端访问的数据的时候示例图提示框无法消失 | | |
| 修复的Bug | a c d e f g h I j l m | 11个 |
| 不能重现的Bug | b | 1个 |
| 这个产品就是这样设计的，不是bug | 无 | 0 |
| 没有能力修复，将来也不打算修复 | 无 | 0 |
| 这个bug的确应该修复，但是没有时间在这个版本修复，延迟到下一个版本修复 | k n o p | 4个 |

1. 场景测试

小杨同学现在有一段医学指南中的内容，想要将这篇文章输入到系统中，获得这段文字的结构信息以及各个段落中存在的相关联的病症信息

小李同学现在有一篇医学指南的内容，需要将这篇文章输入到系统中，得到这篇文章的层次结构以及层次联系清晰的知识图谱

1. 测试矩阵

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试功能 | 测试项 | 检验点 | 预期结果 | 火狐 | Chrome | QQ浏览器 | Safari | Opera |
| 首页功能 | 提交部分 | 用户输入文章内容 | 正确获取用户输入 |  |  |  |  |  |
| 用户重置输入内容 | 正确清除内容 |  |  |  |  |  |
| 用户提交文章 | 提交正确则有返回，失败则进行提示 |  |  |  |  |  |
| 显示部分 | 显示返回结果内容 | 正确显示返回的知识图谱 |  |  |  |  |  |
| 显示结果节点可以拖动 | 显示的知识图谱中的节点可以拖动 |  |  |  |  |  |

1. 出口条件

###### 系统运行环境

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 具体说明 | 时间需求 | 空间需求 |
| 服务器运行环境 | linux以及windows主流环境上可正常运行 | 无 | 无 |
| 数据库服务器 | linux以及windows主流环境上可正常运行 | 无 | 无 |
| 前端运行环境 | chrome，火狐，IE9以上等主流浏览器上可以正常显示 | 无 | 无 |
| 服务器负载 | 能满足1000人同时访问 | 不足1000人同时访问时，平均等待相应时间不超过5s | 无 |
| 服务器稳定性 | 能保证在7\*24h的运行环境下维持项目运转 | 无 | 无 |

###### 系统界面显示

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 具体说明 | 时间需求 | 空间需求 |
| 分辨率适配 | 在不同屏幕分辨率的情况下能正确显示页面 | 无 | 无 |
| 字体适配 | 在不同主流浏览器中运行时能保证正确显示字体 | 无 | 无 |
| 分析结果展示插件显示 | chrome，火狐，IE9以上等主流浏览器上可以正常显示结果 | 无 | 无 |
| 分析结果展示适配 | 对于复杂的本体分析结果要具有一定的适配性，保证本体显示的清晰 | 无 | 无 |

###### 系统功能需求

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | 具体说明 | 时间需求 | 空间需求 |
| 医学指南来源 | 目前能针对《心血管疾病诊疗指南》这本书全文进行知识图谱分析 | 无 | 无 |
| 本体库构建 | 目前能针对《心血管疾病诊疗指南》这本书全文进行本体库的构建，构建结果与文章标题段落成对应关系 | 无 | 无 |
| 后台处理时间效率 | 后台应具有一定的时间处理效率 | 对于1000字的文章，对于主流配置机器的处理时间在10s以内 | 无 |
| 后台处理空间效率 | 后台应具有一定的空间处理效率 | 无 | 对于1000字的文章，对于主流配置机器的处理时使用内存控制在500M以内 |
| 词法分析匹配本体库准确性 | 针对一段文字中提取属于本体库内容的准确性应保证在80%以上 | 无 | 无 |

## 三、alpha版本发布说明

1. 本版本的功能

* 可供用户进行文章录入、清空文章录入、显示医学指南分析结果的界面，用户可以将医学指南的内容通过该界面进行提交
* 可以将用户提交的医学指南文章进行分析处理
* 可以访问本体数据库进行医学指南分析
* 整合了THULAC词法分析工具进行词法分析
* 可以对医学指南的正则表达式分析，获取医学指南的层次化信息
* 用户可以在界面上看到分析的树状结果

1. 功能的图片、视频展示

<功能截图.docx>

<视频内容/用户提交文章内容.mp4>

<视频内容/用户点击重置按钮.mp4>

<视频内容/分析结果展示模块展示结果.mp4>

<视频内容/结果可拖动查看.mp4>

1. 运行环境要求

|  |  |
| --- | --- |
| 项目 | 具体说明 |
| 服务器运行环境 | linux以及windows主流环境上可正常运行 |
| 数据库服务器 | linux以及windows主流环境上可正常运行 |
| 前端运行环境 | chrome，火狐，IE9以上等主流浏览器上可以正常显示 |

1. 安装方法

用户直接使用目前市面上chrome，火狐，IE9以上等主流浏览器访问网址即可使用

1. 系统已知问题和限制

* 本体库构建并不完善，本题库需要有足够完善的结构与结果，该项目中本体库的构建是依赖于文章本身，具有一定的局限性
* 分词分解工具THULAC运行速度较慢，对于400字左右的文章需要约8秒时间进行分词分解，需要优化THULAC模块以及本项目
* 数据库访问速度较慢，对于数据库查找本体内容时过慢
* 项目中存在临时文件作为正则表达式的输入，可以尝试不使用中间文件而直接使用文章内容
* 项目中存在内存泄漏的问题，初步判断是在THULAC分词工具和数据库访问中存在资源泄漏
* 前端界面在遇到较多较复杂的返回结果时显示不够清晰

1. 发布（并未实际发布）

###### 发布方式

将项目部署在阿里云服务器上



###### 发布地址

初步使用服务器外网地址作为访问入口，后期可以申请域名

## 四、小结

1. 源码仓库地址

https://github.com/Tustune/MedicalKnowledgeMappingTool

1. 过程回顾

###### 项目预期

|  |  |
| --- | --- |
| 团队项目的目标 | 1. 实现一个可供用户进行文章录入、清空文章录入、显示医学指南分析结果的界面，用户可以将医学指南的内容通过该界面进行提交  2. 实现一个可进行医学指南文章处理的后端服务器，用于将用户提交的医学指南文章进行分析处理  3. 实现一个可以进行医学指南分析的本体数据库  4. 实现对THULAC词法分析工具的项目整合，用户进行词法分析  5. 实现对医学指南的正则表达式分析，获取医学指南的层次化信息 |
| 预期的典型用户 | XXX是某医院某科室的一名临床医生，年龄30岁，这类用户所占的市场比例达到70%，属于非常重要的主体用户。XXX需要在医院或者家中进行医学知识温习与拓展的时候会经常性使用医学指南这类书籍进行医学知识的查找与寻找相关医学知识的联系。XXX由于是临床医生，平时生活作息较为规律，由于医学领域学习范围十分广泛而且知识点之间的联系复杂，所以他会经常性的进行医学指南的查阅来巩固与提升自己的医学水平。同时作为医学专业出身，他能够比较熟悉医学指南的结构信息与医学指南的行文形式。但是，有时候对于一些罕见病例或者与自己领域交集较少但是还是有联系的一些领域，XXX常常需要进行相关内容的医学指南查阅，但是医学指南不能直观的反映病症的关联关系，不能快速的查询到关键的知识点，所以XXX倾向于使用一种医学图谱的生成工具，从而清楚明白的显示当前病症的相关信息以及该病症与其他病症之间的联系。 |
| 预期的功能描述 | 1. 用户可以将文档内容粘贴到项目提交框中提交 2. 用户在点击生成医学图谱之后可以获取到系统反馈的医学图谱 3. 系统可以提取文本文件中的格式信息 4. 系统可以调用实体库来判断当前文本文件中的信息关系 5. 系统可以生成实体类型的病症信息，并在前端展示医学图谱 6. 系统采用REST框架搭建java web项目，前后端分离 7. 用户提交文本文档时应该确保提交的稳定性，确保每次提交文档的时候不出现崩溃现象，以及提交无响应现象 8. 系统代码应多写注释和文档，确保系统在维护时有充分的可参考信息，并且代码可读性强 9. 用户在上传文档之后应该在5秒中的时间内生成医学图谱并显示出来 10. 本系统采用REST风格编码，基于B/S模式从而能在不同的平台上运行 |
| 预期的用户数量 | 日均100人 |

###### 满足的需求

* 医学指南按照标题分析成结构化信息
* 构造本体库存放本体之间的联系
* 医学指南在本体库中查找相关内容
* 图形化的显示结果

###### 一次迭代完成的目标

* 前端界面接受用户的输入，能够获取到文章的标题与正文的内容，本版本中文章标题没有实际作用，主要是以正文内容作为处理
* 前端界面重置用户输入，将用户的输入内容清空
* 前端界面供用户进行文章提交，提交到后台进行数据处理
* 后端服务器获取到用户的输入，将用户提交的文章存储为临时文件，供正则表达式模块使用
* 后端进行正则表达式处理，将原文章按照标题关键字进行文章分段处理存储成为层次化信息，将处理结果存储成对应的对象集合
* 后端整合THULAC词法分析工具，可以进行分词分解处理，将正则表达式模块提供的对象集合中的内容进行分词分解，分解成分词的字符串集合
* 后端进行本体数据库访问，将分词分解模块分解成的字符串集合进行组合，将每种组合方式进入到数据库中查找是否存在对应内容
* 后端将生成的本体集合转化成json格式提供给前端进行解析
* 前端界面显示后端提供的json格式的结果，并解析生成图形化的分析结果

###### 团队分工协作的经验及教训

团队中有一名PM，四名开发者，一名测试以及两名文档人员。其中PM同时也进行开发工作，共五名开发者，两名前端三名后端并且由后端的人员进行整合。团队之所以这样进行分工协作是有很多原因的。

首先，最重要的一个原因就是技术栈的差距和项目周期短之间的矛盾。如此分工并不意味着测试和文档人员的技术、代码能力很差，而是因为恰好他们的技术栈与本项目需要的技术不相符而已。项目周期短意味着没法进行新技术的学习和培训，因此在此次团队项目中，安排他们进行测试和文档的整理书写再合适不过。

其次，在项目周期这么短的前提下，团队的分工协作、项目的整体架构、团队成员的技术栈这三者之间有着不可分割的联系。原本的需求是做一个桌面应用，然而在成员的技术栈约束下，如果要实现桌面应用的话，要么只能由两名有丰富c/c++编码经验的人来全部写完，要么只能由三名成员用JAVA来全部写完，剩下几名有丰富web经验的大牛根本没有发挥空间。因为这三者之间紧密的联系，我们小组的项目最终决定实现为web项目。

经过这次的团队项目，对于成员分工协作的经验教训有以下几点

* 团队成员的分工协作安排要充分考虑到各个成员的技术栈的差别和重叠之处与本项目需求的关系
* 分工协作时，人与人之间的沟通占了很大一部分时间，因此PM要注意协调人与人之间沟通的时间、内容和关注点。如果能提前写好规范化的文档来管理沟通过程的话会更好一些。
* 在管理整个项目的过程中，团队成员的能够编码的时间也需要考虑在项目进度中。每个人空余时间的差异导致对项目进度管理的难度大，也导致项目完成的质量不高等这些问题。毕竟不是全职的完成这个项目，因此每个成员的时间管理及安排成了此次迭代过程完成质量的决定性因素。

###### 项目管理的方法

在Github上管理项目的版本。由于大家的时间安排非常的参差不齐，因此我们的完成冲刺过程的时候有些部分采用的瀑布模式进行的开发。即有些部分开发完成之后，直接交予后面模块的领取人进行整合，因此我们并没有用文档或者工具进行项目管理。自始自终都是整合的人李泰郎在管理着代码，代码的版本则交由Gighub来管理。

###### 时间/质量/资源的平衡

通过对团队成员的有效时间和技术栈的平衡，最终项目的整体安排为：10月27日至10月31日安排一次敏捷冲刺；11月1日至5日进行各种测试、BUG的修改、文档的组织完善；11月6日完成PPT内容和所有的文档。

因为团队的成员李泰郎、杨璇和陈健、乔雨相继出差，所以alpha版本完成的质量偏低，有几个bug需要在发布之前修正。比如前端显示问题，分词部分内存泄漏的问题。即使有些bug更改起来不难，但是在之前的迭代过程中实在没时间来修正，因此遗留到下一版本来解决。

1. 团队项目的实际进展

在此次迭代中，因为交叉出差，所有的模块完成度比计划要提前许多。首先因为陈健在迭代期间只有27日、28日、29日三天有时间，所以前端的5个模块都压缩在这三天完成，在燃尽图上就反映出了实际比计划要快的事实。

其次，因为李泰郎和杨璇10月30日开始出差，且代码遇到了几个严重的bug因此30日的后端任务完成量变少。最后，由于冲刺压力，最终在31日完成了此次冲刺任务。从实际花费的时间上也能看出来，29日及之前大家在完成模块过程的时间分配比较充足，但是30日之后就会略微平滑一些。

整体上来说，此次燃尽图较为真实的反映出了此次团队项目的进展状况

1. 团队成员的角色和具体贡献

| **名字** | **角色** | **团队贡献分** | **可验证的贡献** |
| --- | --- | --- | --- |
| 杨璇 | PM | 65 | 权衡时间安排敏捷冲刺、迭代过程的模块分解、敏捷冲刺每日报告、燃尽图绘制、参与冲刺过程并领取完成3个模块的任务、参与完成项目整体的测试、参与完成《团队项目总结》，完成《系统设计说明书》 |
| 李泰郎 | Dev | 60 | 参与冲刺过程并领取完成5个模块的任务、完成测试过程的后端用例测试、参与完成《团队项目总结》、参与完成《系统设计说明书》 |
| 陈健 | Dev | 45 | 参与冲刺过程并领取完成5个模块的任务、参与完成《系统设计说明书》 |
| 乔雨 | Dev | 40 | 参与冲刺过程并领取完成2个模块的任务 |
| 王涵 | Test | 35 | 参与测试过程的前端用例测试，演示PPT |
| 万珊红 | Doc | 35 | 准备用于测试的格式化文章、参与完成《测试报告》，参与总结bug，写PPT |
| 蒋文佳 | Dev | 35 | 参与冲刺过程并领取完成1个模块的任务 |
| 吴宗瑞 | Doc | 20 | 参与完成测试矩阵，参与完成《测试报告》 |

1. 下一阶段的展望

* 本体库构建并不完善，本题库需要有足够完善的结构与结果
* 分词分解工具THULAC运行速度较慢，对于400字左右的文章需要约8秒时间进行分词分解，需要优化THULAC模块以及本项目
* 数据库访问速度较慢，对于数据库查找本体内容时过慢，需要优化数据库访问
* 项目中存在临时文件作为正则表达式的输入，可以尝试不使用中间文件而直接使用问文章内容
* 项目中存在内存泄漏的问题，初步判断是在THULAC分词工具和数据库访问中存在资源泄漏，需要进行内存泄漏的控制与检测
* 前端界面在遇到较多较复杂的返回结果时显示不够清晰，需要重新设计一下页面的显示