小句复合体语料标注系统设计与实现

指导老师： 罗智勇

学号： 2015411680615

学生： 周玥

**【摘要】随着互联网整个行业的快速发展，前后端分离开发已经成了当下的新趋势。本研究致力于将文本标注工作线上化，于是便采用前后端分离思想开发了小句复合体语料标注系统。该系统简化了文本标注过程，并且采用数据库储存标注数据，减轻了许多繁琐的文件整理工作，为研究小句复合体的师生带来了一定的便利。**

【**关键词**】 小句复合体; 标注; 前端; 数据库

**【Abstract】**With the rapid development of the entire Internet industry, front-end and back-end separation development has become a new trend. This study was devoted to the onlineization of text annotation work. Therefore, a NTC marking system was developed using front-end and back-end separation concepts. The system simplifies the text annotation process, and uses the database to store the annotation data, which relieves a lot of cumbersome document organization and brings convenience to the teachers and students of studing NTC.

【**Key Words**】 NTC; Label; FED; Database

1. **引言**

小句复合体（NTC）语料标注是一项复杂的系统工程。按照标注规范，生语料需要人工对每一个标点句进行基于图的标注，同时还要按照转换规则为标记语料生成缩进格式、话题自足句等形式。目前，我们对文本进行标注工作时，往往都是人工借助Word等文档工具通过手动缩进或者标记来达到标注效果。但这种通过Word来做标注工作的方式太过于繁琐，大量的缩进与标记将会花费标注者的大量时间和精力，而且容易犯错。

因此，开发一个基于WEB端的文本标注系统，通过友好的人机交互界面，使标注工作者能用更轻松以及更方便的方式、甚至采用众包等协同工作的方式，来进行文本标注工作，从而提高工作效率。这对小句复合体项目的研究具有重要意义。

* 1. VUE简介

Vue是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

* 1. JQuery简介

jQuery是一个快速、小型、功能丰富的JavaScript库。它使HTML文档遍历和操作、事件处理、动画和Ajax等变得更为简单，使用易于使用的API，可以跨多个浏览器工作。由于通用性和可扩展性的结合，jQuery已经改变了数百万人编写JavaScript的方式。

* 1. MYSQL简介

Mysql是最流行的关系型数据库管理系统，在WEB应用方面MySQL是最好的RDBMS(Relational Database Management System：关系数据库管理系统)应用软件之一。

1. 系统分析
2. 需求分析

通过分析，小句复合题语料标注系统应具有以下模块：

1. 用户登录、退出登录页
2. 标注语料以及标注语句选择页面
3. 标注工作页面
4. 用户管理页
5. 可行性分析

由于用户在标注语料列表页面以及标注语句列表页可以自主选择标注任务，而这样可能会导致一个问题，即当某个用户选择了第一份任务进行标注时，另一个用户此时不知道这个情况，同样也选择了这份任务进行标注。这样就会导致很大程度的重复劳动。

因此，只是简单的实现选择列表页面是不可行的。在经过一番研究与咨询后，决定在系统中加入一个权限限制功能，即某个用户选择一个语料之后，系统就将这个语料锁定，只允许当前用户查看以及标注，这样，其他用户就看不到这个语料，除了拥有最高权限的用户。而且用户管理页面同样只允许最高权限用户进入。

1. 系统设计
2. 系统功能结构

1. 开发及运行环境

前端框架以及IDE：Vue.js、WebStorm

后端语言以及编辑器：Python、Visual Studio Code

数据库：MySQL

Web服务器：Internet Information Services（IIS）

1. 数据库设计

以下三张列表是该系统的数据库结构设计

表3-1用户信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colume Name | Primary Key | DataType | Length | Auto Inc | Comment |
| id | 是 | VARCHAR | 45 | -- | 用户id |
| name | -- | VARCHAR | 45 | -- | 用户名 |
| password | -- | VARCHAR | 45 | -- | 用户密码 |
| root | -- | INT | 1 | -- | 用户权限 |

表3-2 语料以及语句信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colume Name | Primary Key | DataType | Length | Auto Inc | Comment |
| id | 是 | INT | 11 | 是 | 数据id |
| data | -- | VARCHAR | 10000 | -- | 数据 |
| isFinished | -- | TYNYINT | 4 | -- | 是否完成 |
| currentUserId | -- | VARCHAR | 45 | -- | 当前用户 |
| parentId | -- | INT | 11 | -- | 父id |

表3-3完成数据信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colume Name | Primary Key | DataType | Length | Auto Inc | Comment |
| id | 是 | INT | 11 | 是 | 数据id |
| data | -- | VARCHAR | 10000 | -- | 数据 |
| lastUserId | -- | INT | 11 | -- | 最后用户 |
| originId | -- | INT | 11 | -- | 原数据id |
| parentId | -- | INT | 11 | -- | 父id |

1. 系统总体架构

（一）、系统登录页面

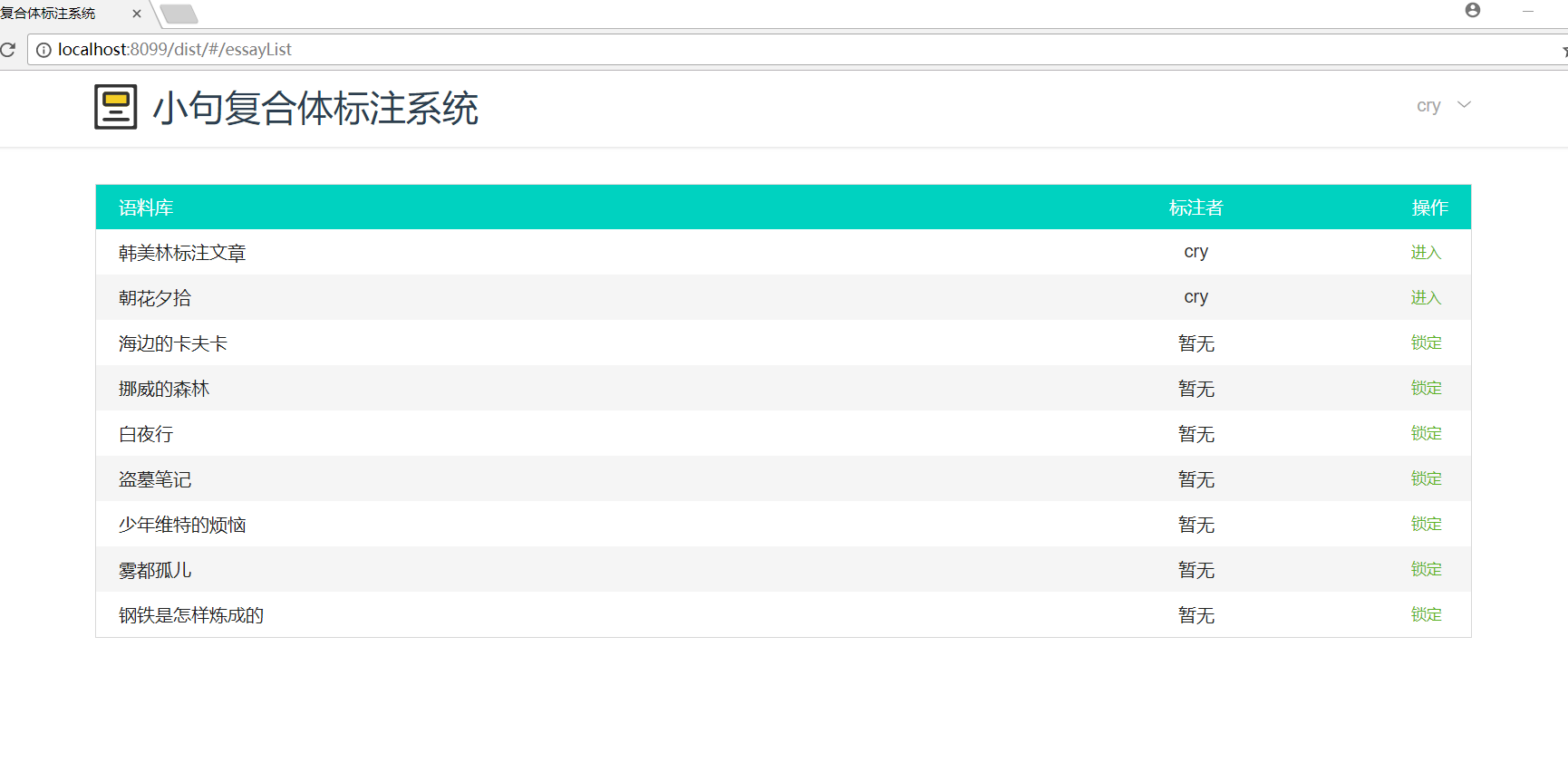
系统首先进入的是登录页面，如图4-1所示

图4-1 登录页面



（二）、系统首页-语料列表页 如图4-2所示

图4-2 语料列表页



（三）、选择语料后-语句列表页 （预览功能图4-3-1） 如图4-3所示

图4-3 语句列表页



图4-3-1



（四）、选择语句后-标注页（以及修改标注页图4-5） 如图4-4所示

图4-4 标注页



图4-5 修改标注页



（五）、标注页面-标注菜单 如图4-6-1，4-6-2所示

图4-6-1

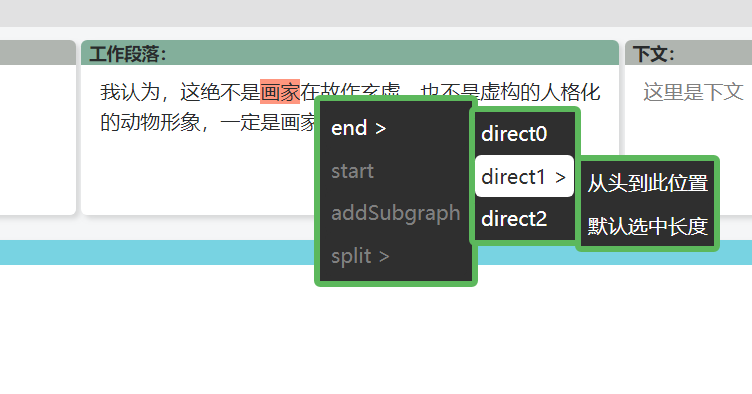
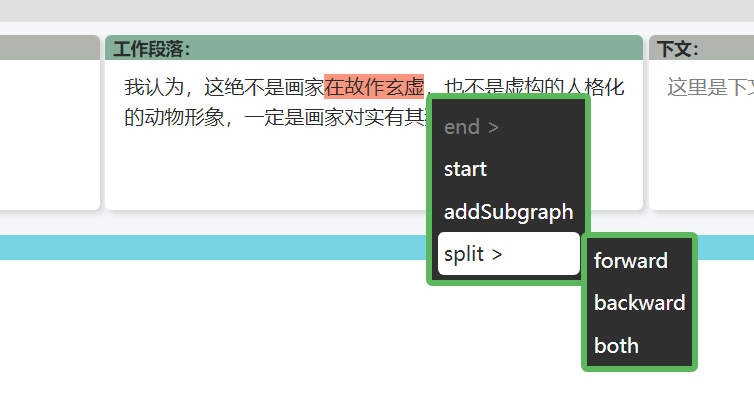


图4-6-2



（六）、标注页面-预览区域 如图4-7-1，4-7-2所示

图4-7-1

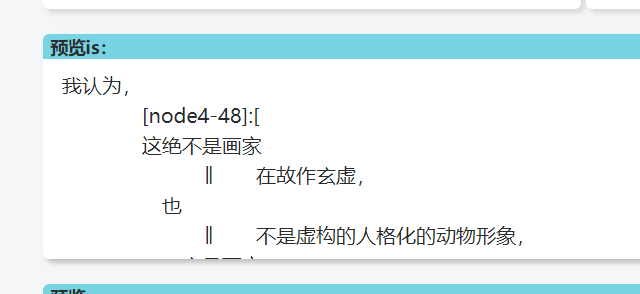
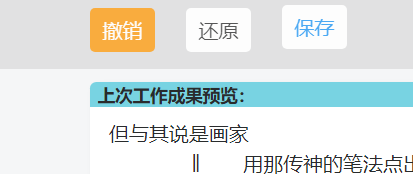
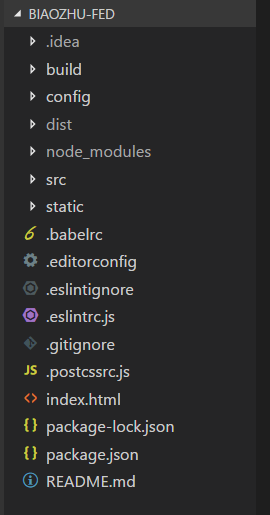


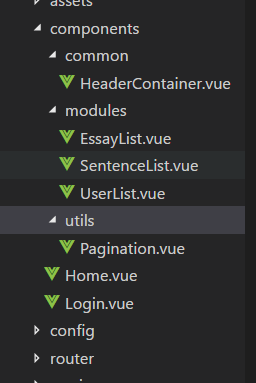
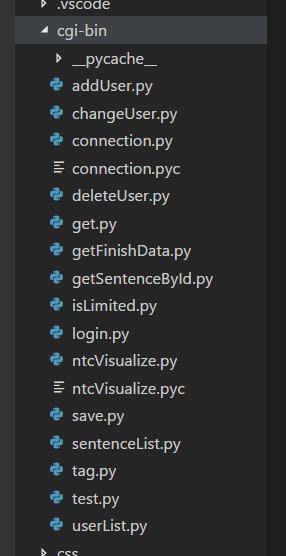
图4-7-2



（七）、标注页面-撤销、还原、保存按钮组 如图4-8所示







1. 系统主要功能模块设计

（一）、登陆模块设计以及代码分析

首先，系统的用户是由用户管理后台分配的，用户拿到用户名以及密码就可以进入登陆页面输入正确的用户名以及密码就可以成功进入系统。但当用户输入不存在的用户名或者错误的密码时，系统会给予用户对应的错误提示。

实现代码如下：

1. 前端JavaScript代码
2. 登陆方法：

前端检验用户名或密码是否为空，如果为空，则不发送请求，并提示用户。

login () {

if (this.loginInfo.name && this.loginInfo.password) {

this.isNull = false

loginAxios(this.loginInfo, this.afterLogin)

} else {

this.isWrongInfo = true

this.wrongInfo = '密码或用户名为空，请输入'

}

}

1. 进入检查是否已经登陆

checkLoginInfo () {

const userName = getCookie('userName')

if (userName) {

this.$router.replace('essayList')

} else {

const name = getCookie('userName')

const password = getCookie('password')

if (name) {

this.loginInfo.name = name

}

if (password) {

this.loginInfo.password = Base64.decode(password)

this.isRemember = true

}

}

},

afterLogin (res) {

if (res) {

if (res.success) {

const user = res.data

setCookie('userName', user.name)

setCookie('userId', user.userId)

setCookie('root', user.root ? user.root : '')

if (this.isRemember) {

const esPassword = Base64.encode(this.loginInfo.password)

setCookie('password', esPassword, 10)

} else {

delCookie('password')

}

this.$router.push('essayList')

} else {

this.isWrongInfo = true

this.wrongInfo = res.message

}

} else {

this.isWrongInfo = true

this.wrongInfo = '未知错误!'

}

}

},

created () {

this.checkLoginInfo()

}

1、设计核心标注页面的交互方案。

2、确定实现标注功能所需要的文本位置信息，以及处理后的数据结构。

3、按照设计方案用前端描绘页面整体结构，其中设置用户工作文本区域，用鼠标选取文本的方式确定选中文本位置信息。（该步骤中，主要研究前端Selection这一Web接口）

4、前端通过ajax将用户标注的位置信息传递给python脚本（通过CGI接受请求并返回计算结果），然后前端将返回数据中完成标记后的文本通过把换行符与空格缩进等格式标记符号替换成对应的html标签，最后将标记后的句子结构展示在用户面前。用户通过一个相对简单的操作就能直观看到对应的标注结果。

5、核心标注页面完成后，建数据库（MySql）将现有的待处理文本通过脚本批量存储到数据库里。

6、设计用户登录页面以及待处理文本列表页，与标注页面连接。

7、前后端对接实现整个标注系统的大流程。

8、用户权限设置等细节功能的设计与实现。

s