本科生软件课设报告

题　　目：在线事件标注工具设计与开发

院　　系　电子信息与通信学院

专业班级　电信 1705 班

姓　　名　杜咏琦

学　　号　U201713356

指导教师　王邦

2020 年 01 月 15

一．项目描述

随着 NLP 的不断发展，把无结构化的原始文本结构化，以获得各项 NLP 任务的预料标注就越来越重要。目前已经有很多比较成熟的标注工具，比如 BRAT，基于 web 的文本标注工具，还有 TEEDA、Chinese-Annotator 等标注工具。而本项目的目标就是搭建一个能对原始文本标注并能导出最终标注结果的标注工具。

本项目中的标注工具最终实现的功能包括:

1.普通用户的注册与登录功能，文本标注功能，以及对标注后生成 xml 文档的下载功能

2.用户组长的登录功能，上传文本功能，确定最终标注功能以及对原始文本和标注文本的下载功能。

3.管理员可进行小组的创建，组长的任命等对数据库的直接操作。

具体实现功能如下图所示。

二．系统描述

本系统在window10操作系统中进行部署,使用Django内置的轻量WSGI服务器做为web服务器,数据库管理选择sqlite3。

1）系统开发环境

1.OS环境

Windows操作系统,Windows10专业版

2.硬件环境

（一般配置）

CPU： 2.90GHz,双核，四逻辑处理器

内存：8GB

浏览器：谷歌，IE

显示分辨率：1920*1080

3.web框架

Django框架,版本:3.0.1

4.参与开发软件

软件名称	使用语言
Pycharm	Python3.7
bootstrap	H5+css+jQuery

2） 系统架构

本系统开发采用MVT架构，即model-view-template架构，model指数据库映射部分，view指根据请求进行相应数据库操作并返回对应结果与操作的函数块，template指网页的显示部分。

总结说来即template层用来显示,根据路由可以调用view层对应的函数，这个函数可以根据请求与数据对数据库model层进行相应的操作，最终将相应结果返回到templete层显示出来。

3）版本控制

使用git实现远程仓库管理与版本控制，并使用github实现多人协作。

4）启动命令

激活conda虚拟环境并安装Django包后,在命令行输入py manage.py runserver

5）关闭命令

命令行停止执行 runserver 命令。

三．模型描述（这一部分介绍数据库的结构）

有关本项目中的数据库结构，即 model 部分，在 django 框架中，可通过在 modelss.py 文件中定义类来实现对数据模型（数据库表）的定义（而对数据库的增删查改等操作在 view 层实现）。

本项目中被定义的类包括四个，即用户 myUser，组 group，初始文档 text 以及标注后的文档 a_text,其中用户类做为 django 内置 User 类的子类，继承了 User 类的一些定义的属性，如姓名，密码。

•用户 myuser 类所用到的属性包括姓名 name,密码 password,小组 group_id(,foreign key,外键关联 group)，身份 limits.

	id	group_id	user_id ▲	limits
1	14	1	27	1
2	15	1	28	0
3	16	1	29	0

• 组 group 类包括小组成员数 member_number 和小组名称 group_name,id 为 primary key.

	id	member_number	group_name
1	1	3	1
2	2	3	2

• 初始文档 text 类包括文档名称 name，文档物理相对地址 text，文档被标记次数 index(同一用户标记记一次)（标记次数等于小组普通组员个数是才对其计算一致性），文档是否能被组员标注 limit（组长确定生成最终标注后，或一致性到百分之百后，组员不得继续标注），小组 group（外键关联 group）

	id	name	text	index	limit	group_id
1	46	bb	text/1/bb	2	1	1
2	47	t001.txt	text/1/t001.txt	2	1	1
3	48	register.html	text/1/register.html	0	1	1
4	49	learn.html	text/1/learn.html	0	1	1
5	50	aa.txt	text/1/aa.txt	2	1	1

标注后的文档 a_text 类包括标记后文档名称，小组 group（外键关联组 group 类）,用户 user_id（区别同一组不同用户对同一文本的标注，以便计算出 text 类中文本的标记次

数），标记后生成的 xml 的存放**物理相对地址** xml（设计成按组别以及用户名分文件夹的形式）

	id	name	group_id	user_id	xml
1	92	bb.xml	1	16	xml/1/qenqqenq/bb.xml
2	94	bb.xml	1	15	xml/1/jianjian/bb.xml
3	95	t001.xml	1	15	xml/1/jianjian/t001.xml
4	96	t001.xml	1	16	xml/1/qenqqenq/t001.xml
5	97	aa.xml	1	16	xml/1/qenqqenq/aa.xml
6	99	aa.xml	1	15	xml/1/jianjian/aa.xml

四．软件设计

4.1 软件前端最终呈现效果设计

从软件的呈现效果来看，软件包括：

1.注册界面与登录界面。

2.用户的文本选择界面与标注界面（包括导航功能）。

3.用户组长的文本上传页面，文本筛选界面和可打开的两个组员的标注界面（包括导航功能）。

4.管理员的登录界面与管理界面（包括导航功能）。

4.2 软件前后端代码设计过程

整个软件设计过程为需求型设计，即根据即将实现的功能一步一步推展进程

当然首先是确定系统的架构为 MVT,在确定并搭建好 web 框架后，开始进行需求分析与软件的完善。

该项目共有三种类型用户，包括普通用户，组长和管理员。

对于普通用户来说，普通用户可在注册登录界面进行注册并选择加入相应的标定小组，注册后跳转至登录页面进行登录，普通用户登录后拥有文本选择与标注的权限，并可对标注后的文本进行下载。

对于用户组长来说，用户组长在经登录页面登录后，具有上传文本以及确定最终标注的权限，同时为避免文本上传重复，用户组长还具有查看已经上传文档的权限。

而对于管理员来说，管理员可直接对后台数据库进行操作。

4.21 后端代码设计过程

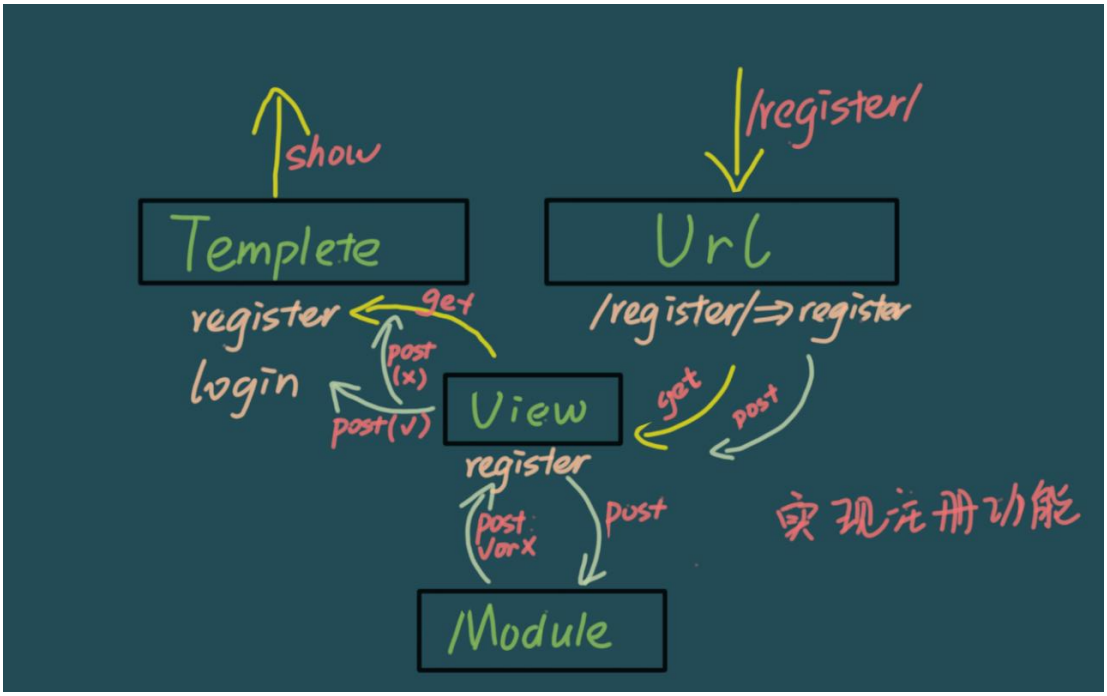
后端整个设计过程大致如下图所示。(根据功能的设计流程)(画图时没注意有单词写错了,图里面的 templete 应该改为 template,module 应该改成 model)

下面得所有图均显示了从浏览器输入路由请求网页或网页提交之后经过 view 中函数得处理,处理包括操作 model 数据库部分,以及渲染 template 部分生成 html 页面在浏览器上显示的过程.

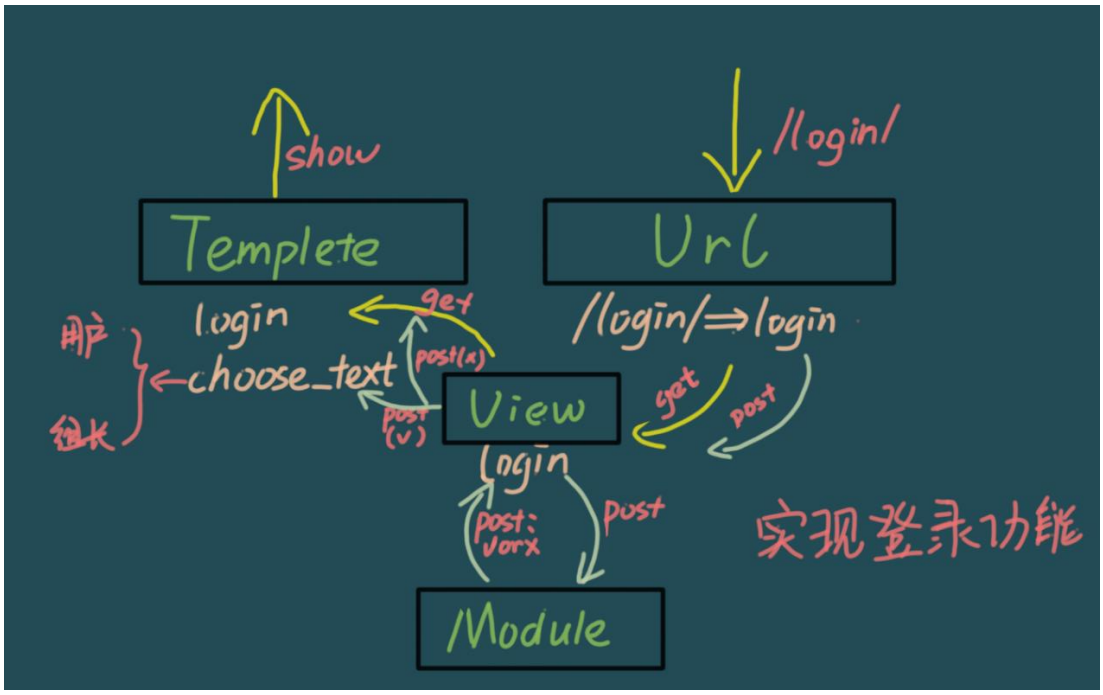
注意每个图都包括浏览器初始请求页面的 get 请求(**黄色箭头**)和浏览器提交数据或请求的 post 请求(**灰色箭头**)两条路径,两条路径用两种颜色标识出来了.

所有同种颜色的箭头组成一条从输入路由到显示的路径.

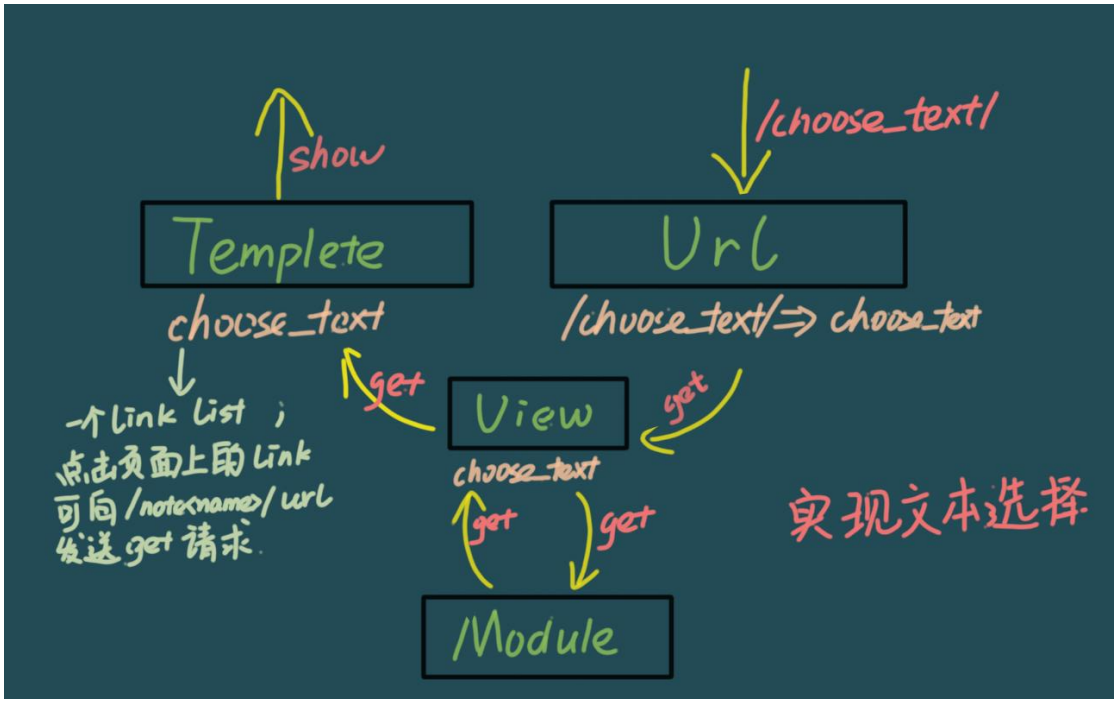
1. 实现注册功能



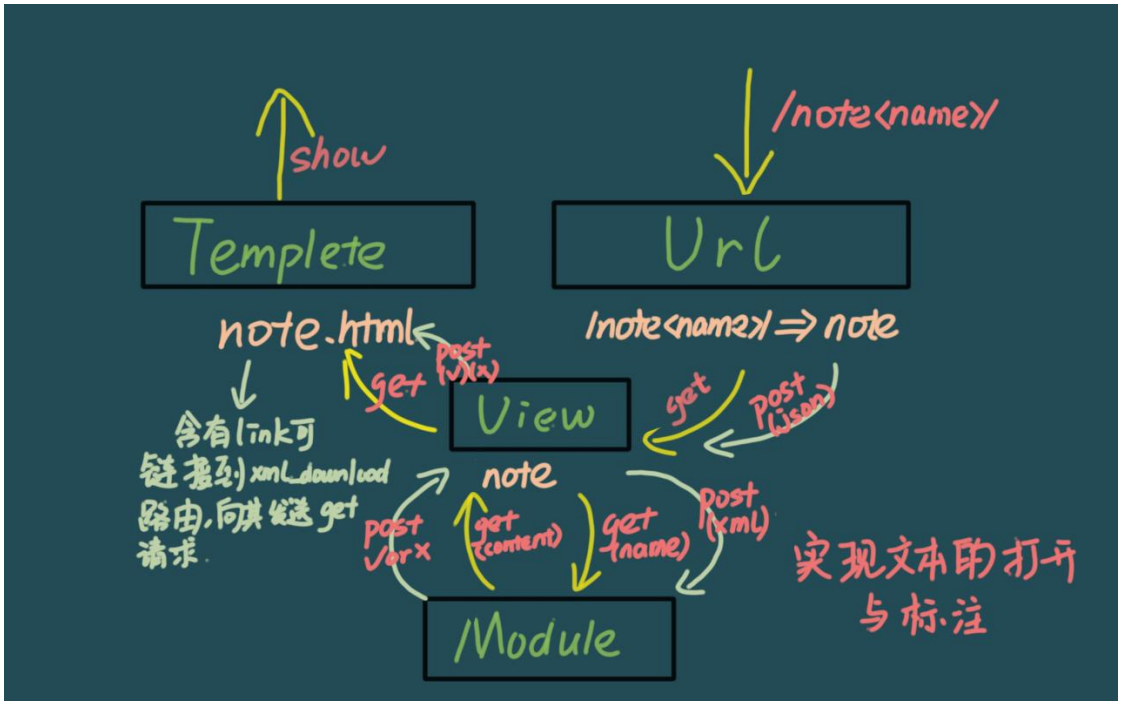
2.实现登录功能


show
/login/
Templete
Url
login
choose_text
/login/⇒login
get
post(x)
post(√)
View
get
post
login
post:
vorx
post
/Module
实现登录功能

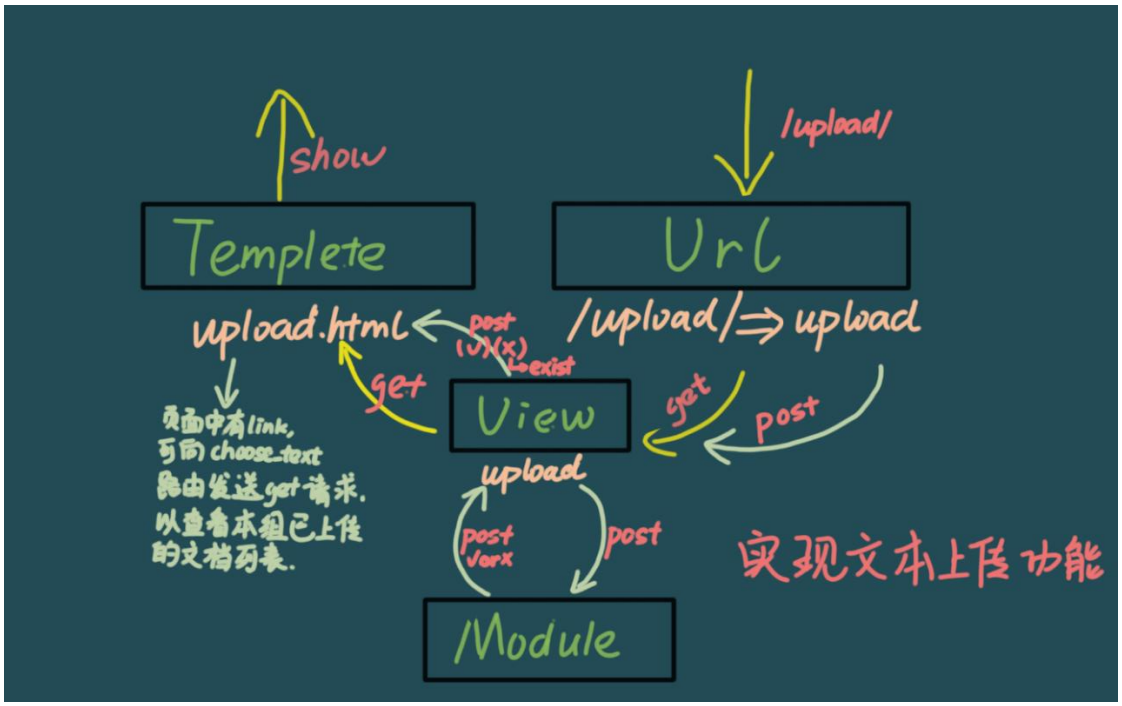
3. 实现文本选择功能


show
/choose_text/
Templete
Url
choose_text
/choose_text/⇒ choose_text
get
View
get
choose_text
get
get
/Module
实现文本选择

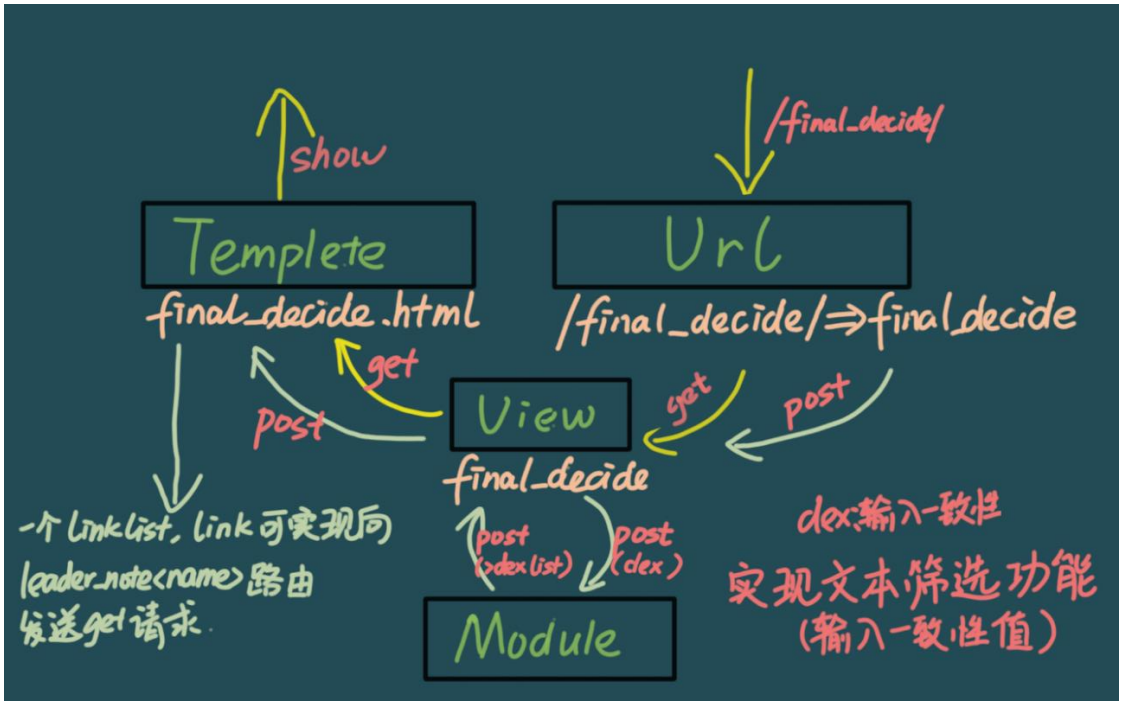
4. 实现普通用户标注功能



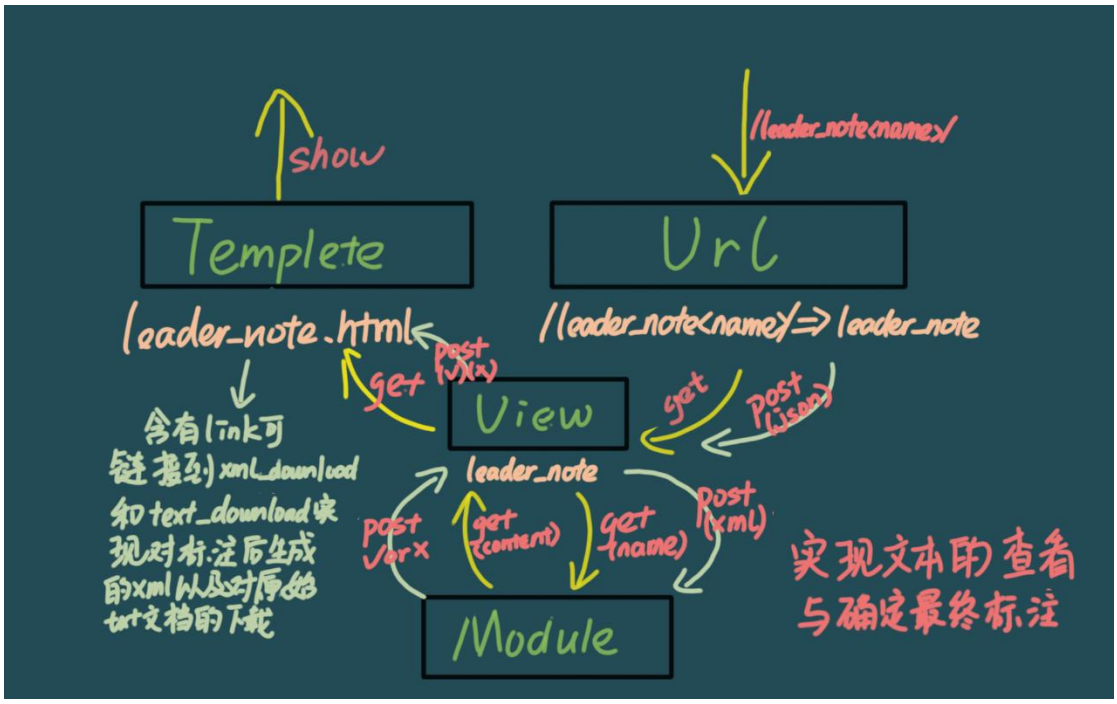
5. 实现用户组长的文本上传功能



6. 实现用户组长的文本筛选功能



7. 实现用户组长的标注功能（因为要实现组长可查看组内两个人的标注，因此除此页面路由外另外加了一个页面 leader_note1 显示另一个人的标注，下图所示页面中有链接可来链接到另一个页面）

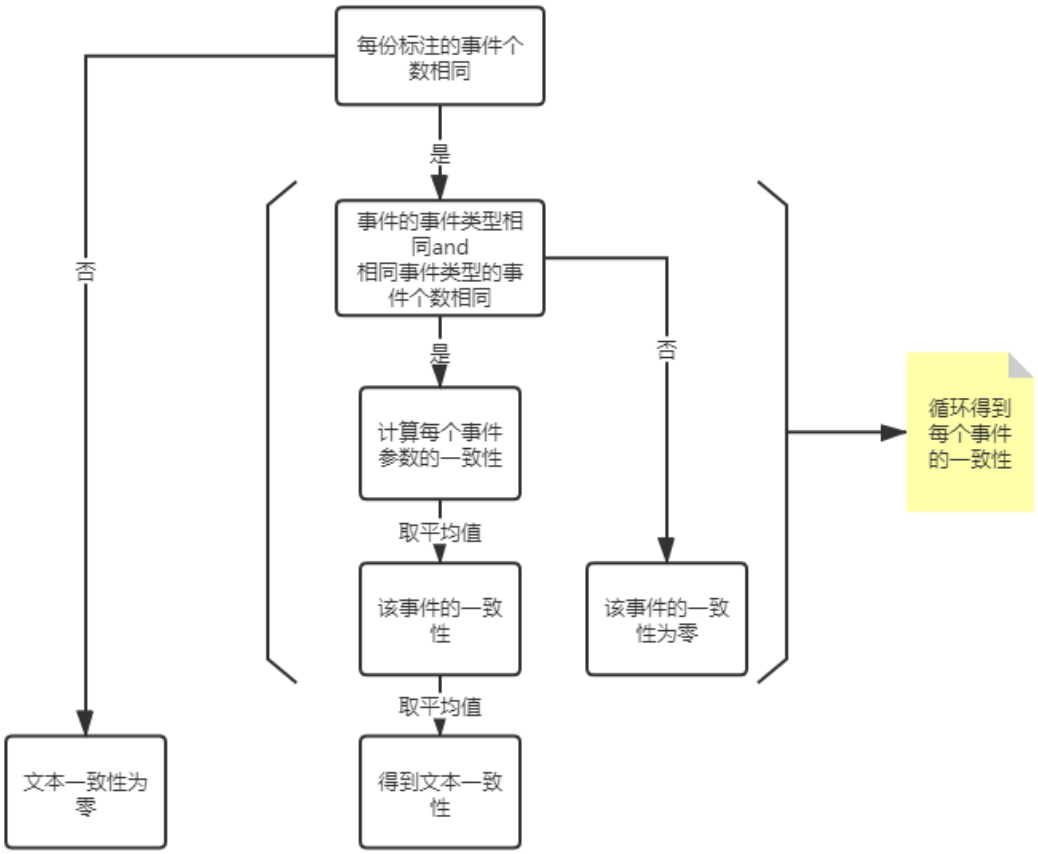


8. 一致性计算的实现

对于一致性的计算，我们设计的判断条件是在组员全部标记后才开始进行一致性计算，计算的方法是，首先判断每个人标记的事件个数是否相同，如果组员标注的事件个数有差，则总的文本的一致性设定为零，接着判断同一个事件里每个人标注的事件类型是否相同，如果组员对同一个事件的事件类型标注有差，则这个事件的一致性为零，接着判断同一个事件类型里组员标注的事件参数个数是否相同，如果不同，则这个事件的一致性为零.

如果上述几个判断都通过，则计算对同一个事件同一个参数的组员的标注的交集的字数 a 与并集的字数 b，用 a/b 表示一个参数的一致性，计算出所有参数的一致性后，取平均值得到这一个事件的一致性，再对其他事件进行相同的操作得到每一个事件的一致性，对所有的事件的一致性取平均值即得到文本的一致性.

流程图见下图：



备注 1:有考虑过如果只有一两个人标注有误而使文本一致性为零的情况,这种情况因为如果有人长期有意标注错误,组长方则会发现待最终确定的文本会持续没有更新,可联系管理员寻找原因,给予警告,这样能从根本上避免每次计算一致性均需要剔除一两个人的标记的情况,因此最终设计为不进行奇异样本的剔除.

备注 2:计算交并集时,根据偏移量计算而非文字内容,可以防止出现文字相同而不是同一个位置的文字造成一致性计算结果的错误.

备注 3:并未给予触发事件更高的权值,而是取平均,因为触发事件的重要性在最初的判断阶段已经给出,如果同一个部分的事件触发事件类型不同,这个事件一致性直接为零,是类型相同与否起绝对作用而非触发事件是什么内容,因此在用交并集来计算一致性的时候,触发事件与其他的事件参数并无区别,因为他们在这个计算步骤里只是文字或者说偏移量.

4.22 前端代码设计过程

主要介绍两个界面的设计，用户标注界面和组长最终标注界面。（这里的用户指组员）

①用户标注界面

对于用户标注界面,用户的标记过程为用鼠标选取一段文字,在页面中点击相应的按键,便可将这段文字标记成一个事件的对应成分,用户标记完成后点击提交即可对生成的标注的 xml 文档进行下载。这个过程经历了，用户选择文字，用户点击屏幕中的某一个按钮，用户标记完毕后的提交。

当用户选择文字后，暂时记录下这段文字相对句首的偏移；

在用户点击相应按键后，给选中的字的两端加上标签，定义标签的一些属性来表示这段文字的偏移值，事件类型，属于哪一个触发事件，是哪一个事件参数；

当用户提交标注后根据这些属性值以及文本的内容，将用户进行过的标注转成 json 的格式传递给后端处理，进而生成标注的 xml 文档。

②组长最终标注界面

组长最终标注界面，组长进行的操作包括查看同一个文本的两个组员用户的标注，并选择一个完成最终的标注。

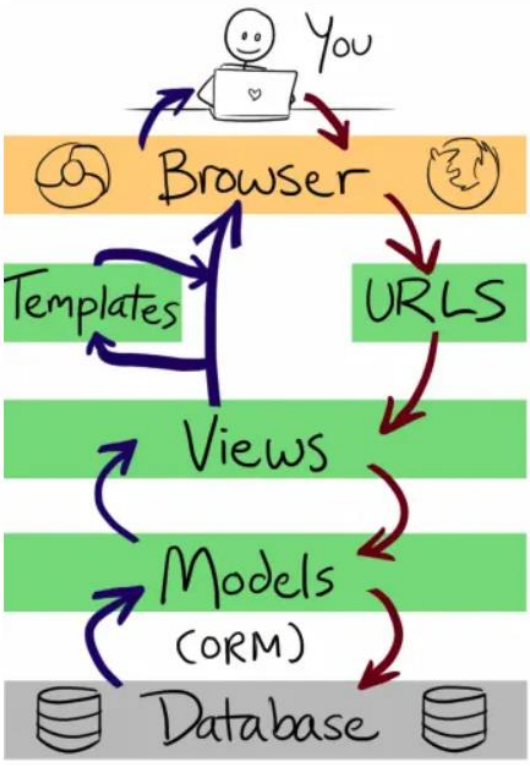
当组长打开查看组员的标注时，页面向后端发送请求，后端传回 json 格式的用户的标注，前端 js 文件中接收 json 数据后，获取文本内容，删除原本的 DOM 树中的文本节点，根据文本内容以及 json 数据中的标注内容，重新构建这一部分 DOM 树的结构，在有标记的部分加上标签，定义属性，（属性与用户标注界面中的标签的属性完全相同）属

性值从 json 数据中获取，无标记的文本部分只单纯做文本节点。这样就实现了 json 数据向 html 中插入。

当前端组长标记时，这时候的前端设计与用户标注界面相同。

另外有关界面布局以及修饰美化部分，html 放在 template 中,其他静态资源,包括 css 与 js 文件放在 static 文件夹中由 template 中的 html 文件引入,另外登陆后界面有导航条的导航条可实现功能跳转,每个 html 页面都有导航条的部分代码,在登陆后会将用户信息传给 template 中的 html,根据用户信息判断导航条应该显示哪些,比如导航条对用户来说包括 choose_text 与 note 两个,对组长来说包括 upload,final_decide,leader_note 和 leader_note1 这四个.

五．模块层次图



具体过程为浏览器向某一路由发送请求，经 urls 文件制定该路由的请求发送给 views 层中的哪一个函数处理，views 层的指定函数处理时，根据发送的请求以及请求中携带的用户信息或一些提交的数据信息，对 model 层定义的数据库进行相应的操作后，通过 migrate 命令将 models 文件的修改迁移到实际的数据库上,将返回的结果传给 template 层中的 html 文件中，再由浏览器显示出来。

六．总结与建议

这次项目里我们组每个人的收获都很大，都花了很多时间，学到了很多东西，当然也碰到很多困难，在整个项目进展过程中最难的阶段我觉得是在最开始的时候，因为脑袋空空，无从下手，在搜索资料的过程中会发现网上的信息太多，各种各样的说法往脑子里灌输，本来以为自己有点思路的时候在接着的信息输入中也变得不确信，最后一片混乱，也就是没法在给出的所有信息中筛选出需要的信息，有时候一整天一点收获也没有。后来开始将中心转到 w3cschool 上的相关前端教程中，对前端相关语言有一些掌握后，开始寻找搭建网站的系列视频教程进行学习，这个时候对文档的存储方式 xml 以及 xml 的解析有了一些了解，因为小组负责后端的两位同学之前都有学习过 python,因此决定采用 django 的 web 框架。因此在视频学习的同时也开始跟着 django 文档的一些基础教程看，这个时候才渐渐有了一点思路，开始上手做这个项目，这个时候才真的有进展，在之前的学习准备过程中整个项目没有丝毫进展确实挺让人心慌，也影响学习效率，总之那段时间我们压力都挺大的，不过也是学习东西最多的阶段。

后来就是每天定要实现的功能，然后我们一起查 django 文档，查看一些博主的博文经验分享，然后结合自己的项目进行代码设计与编写，这个时候的网络搜索效率就很高，因为每次搜索都知道要实现哪一个功能，用什么软件工具，用什么 web 框架，用什么语言，这些清楚后对相关知识的检索效率有很大改善与提升，接着一步一步根据功能需求慢慢完成系统。

总之在整个过程中离不开网页上的相关知识的检索，在大脑里一点相关知识都没有的时候检索很容易出现混乱，所以最好跟着一个比较完整的并且知道之后一定会用到的知识的教程学习，学习后的知识能够帮助筛选有用的信息，也就是不可以试图一次检索就能得到想要的结果，检索只是辅助工具，要先进行知识储备，慢慢的了解自己真的需要查询些什么，这样才不会浪费大量的时间在不停的搜索上。

另外的一个收获就是认识到了操作的可贵，理论的知识如果离开操作很容易会被忘记，而且操作可以加深对理论的理解。在做项目的过程中我常常发现有时候需要用的东西很久之前就读过，不过因为缺少实践，理解不够深刻，记忆也不够深刻，在真的操作的时候在读一遍才发现原来之前漏了很多重要的东西。

回想起来我们在项目的进展中有不足之处，在进展过程中缺乏与其他组的交流与学习，多多交流，互相给出建议可能有利于加快交流的每个组的进度，因为可能会加快对应当学习

的知识的明确过程。

八．参考资料

Django 文档（官方），

博客网站中的大量经验分享博客，

w3cschool 相关网页前端教程以及 git 教程。

九．附录

附上项目 github 地址：https://github.com/duyongqi/AnnotationTool

后续如果有改进的地方会上传到这个地方