

**软件课程设计中期报告（小组）**

**文本标注系统的设计与开发**

**专业班级**  电信提高2201班

**组内成员**  徐文涛 U202213811

李升臻 U202213801

罗皓研 U202213802

杨家豪 U202213812

**指导教师**  王玉明

**日 期**  2024年11月5日

**2024年11月21日星期四**

目录

[1 项目概述 4](#_Toc183124338)

[1.1 项目名称 4](#_Toc183124339)

[1.2 项目内容 4](#_Toc183124340)

[1.3 项目背景 4](#_Toc183124341)

[1.4 相关技术 5](#_Toc183124342)

[1.4.1 前端开发技术 5](#_Toc183124343)

[1.4.2 后端开发技术 5](#_Toc183124344)

[1.4.3 NLP语言处理技术 6](#_Toc183124345)

[1.5 设计任务与要求 6](#_Toc183124346)

[1.5.1 任务清单 6](#_Toc183124347)

[1.5.2 数据支撑 7](#_Toc183124348)

[2 系统需求分析 7](#_Toc183124349)

[2.1 功能需求 7](#_Toc183124350)

[2.2 性能需求 8](#_Toc183124351)

[2.3 用户需求 8](#_Toc183124352)

[2.4 可视化显示 9](#_Toc183124353)

[3 实施方案论证 9](#_Toc183124354)

[3.1 分词模型与算法 9](#_Toc183124355)

[3.1.1 主流分词模型调研与分析 9](#_Toc183124356)

[3.1.2 综合方案 10](#_Toc183124357)

[3.2 后端语言与架构 11](#_Toc183124358)

[3.2.1 Web服务器的调研与分析 11](#_Toc183124359)

[3.2.2 最终选择方案 13](#_Toc183124360)

[3.3 前端框架 13](#_Toc183124361)

[4 系统架构设计 14](#_Toc183124362)

[4.1 系统技术框架 14](#_Toc183124363)

[4.1.1 后端技术框架 14](#_Toc183124364)

[4.1.2 前端技术框架 15](#_Toc183124365)

[4.2 系统功能框架 15](#_Toc183124366)

[4.2.1 工作流程 15](#_Toc183124367)

[4.2.2 界面设计 15](#_Toc183124368)

[5 课程任务分工 16](#_Toc183124369)

[6 课程进度安排 16](#_Toc183124370)

# 项目概述

## 项目名称

文本数据标注系统

## 项目内容

采用B/S架构设计一个软件系统，实现对非结构化文本数据的标注，如：

1. 分词和关键词标注
2. 命名实体标注（人名、地名、组织机构名、日期时间、专用名词）
3. 关系标注
4. 情感标注（正面、负面、中性）
5. 意图标注
6. 语义标注
7. 事件标注（事件的主体、客体、时间、地点、原因、结果等）

并在网页上实现原始文本、标注信息、统计信息的可视化

## 项目背景

文本数据标注是自然语言处理（NLP）领域的关键步骤，旨在为机器学习模型提供高质量的监督数据。标注涉及为文本赋予语义标签，如分类、实体识别或句法结构，支撑模型的训练和性能优化。随着互联网的发展，大规模数据资源变得易于获取，同时标注任务从早期的专家手工标注转向了众包化和自动化。标注方法从单纯的手工操作发展到半自动化辅助和主动学习技术，显著提高了效率。然而，文本标注仍面临高成本、领域依赖和语言资源不均等挑战。未来，智能化标注工具和自监督学习的普及将进一步降低对标注资源的需求，推动NLP技术的广泛应用。

## 相关技术

### 前端开发技术

1. HTML：（超文本标记语言）用于定义网页的结构和内容，包括文本、图像、链接等元素
2. CSS:（层叠样式表）负责美化网页的外观，如颜色、字体、布局等
3. JavaScript：是一种脚本语言，用于实现网页的交互性和动态功能，如响应用户操作、更新页面内容和与服务器通信。这三者相互配合，共同构建现代化、功能丰富的网页
4. Bootstrap框架：Bootstrap 是一个开源的前端框架，提供了一套预定义的 HTML、CSS 和 JavaScript 组件，用于快速开发响应式、现代化的网站和网页应用

### 后端开发技术

1. git：一个分布式版本控制系统，用于跟踪代码变更并协作管理项目版本
2. Navicat：一款数据库管理工具，提供可视化界面以便管理和操作数据库
3. MySql：一个开源的关系型数据库管理系统，用于存储和管理结构化数据
4. Apache：一个开源的Web服务器软件，用于托管网站并处理HTTP请求
5. Flask：一个轻量级的Python Web框架，用于快速构建和部署Web应用

### NLP语言处理技术

1. **神经网络**：一种模仿人类大脑结构的计算模型，通过层级结构处理复杂的数据模式和关系
2. **PyTorch**：一个开源的深度学习框架，提供动态计算图和灵活的编程接口，适用于研究和生产环境
3. **TensorFlow**：由Google开发的开源深度学习框架，广泛应用于大规模机器学习任务和生产部署
4. **LAC框架**：一个中文自然语言处理工具包，基于深度学习用于中文分词、词性标注和命名实体识别
5. **spaCy**：一个开源的自然语言处理库，专注于高效的文本处理和深度学习模型集成，支持多种语言的NLP任务

## 设计任务与要求

### 任务清单

1. 完成系统需求调研与分析
2. 设计本实现相关功能模块及其可视化操作界面
   1. 完成系统数据库结构设计、系统框架及其后端模块设计
   2. 实现原始文本数据管理、标注信息管理、标注数据字段管理、人工标注任务管理、只能标注算法管理功能
   3. 实现对文本数据的人工标注和自动标注
   4. 实现对标注信息、统计信息的可视化展示
3. 完成系统功能开发和联调测试，并部署上线

### 数据支撑

1. 软件系统支持上传txt文件或csv文件或手动输入
2. 验收时，将给大家一批新的数据，现场测试系统功能和效果

# 系统需求分析

## 功能需求

1. 数据上传与管理：系统应支持用户上传 txt 或 csv 格式的文本文件，并对上传的文件进行有效的管理，包括文件存储、查看、删除等操作。
2. 标注任务管理提供多种标注类型选择，如分词和关键词标注、命名实体标注（人名、地名、组织机构名等）、关系标注、情感标注（正面、负面、中性）、意图标注、语义标注、事件标注（事件的主体、客体、时间、地点、原因、结果等）。用户可根据需求自定义标注任务。
3. 实现人工标注和自动标注功能。对于自动标注，应具备一定的准确性和可调整性；人工标注应提供便捷的操作界面，方便标注人员进行标注工作。
4. 标注信息管理：对标注后的信息进行存储、查询、修改和删除操作，确保标注信息的准确性和完整性。
5. 数据字典管理：维护标注过程中使用的各类数据字典，如命名实体类型字典、情感标签字典等，方便标注人员参考和使用。
6. 可视化展示在网页上展示原始文本，标注信息应以直观的方式呈现，如不同颜色标注不同类型的实体或标签。
7. 提供统计信息的可视化，例如各类标注的数量分布、不同情感倾向的比例等，帮助用户快速了解数据的整体特征

## 性能需求

1. 响应时间：系统应具备较快的响应速度，确保用户在进行数据上传、标注操作以及查看结果时能够及时得到反馈，一般操作的响应时间应控制在 1 - 3 秒以内。
2. 准确性：自动标注算法应具有较高的准确性，尽量减少错误标注的情况。对于人工标注，应提供必要的校验和提示功能，以提高标注的准确性。
3. 稳定性：系统应能够稳定运行，在长时间使用和高并发访问情况下不出现崩溃或异常情况，保证数据的安全性和完整性。

## 用户需求

1. 界面友好性：系统界面设计应简洁明了、易于操作，具有良好的交互性。新用户能够快速上手，熟练用户能够高效完成标注任务
2. 个性化定制：允许用户根据自己的习惯和需求对标注界面、标注类型等进行一定程度的个性化设置。
3. 帮助与文档：提供详细的帮助文档和操作指南，方便用户在使用过程中遇到问题时查阅和解决。

## 可视化显示

1. 设计对初始文本数据的可视化显示
2. 设计对分词标注数据的可视化显示
3. 设计对数据特征的可视化显示

# 实施方案论证

## 分词模型与算法

### 主流分词模型调研与分析

1. 分词和关键词标注
   1. LAC（Lexical Analysis of Chinese）：支持中文分词和关键词提取。
   2. 结巴分词（jieba）：常用的中文分词工具，简单易用。
   3. TF-IDF：用于关键词提取的经典算法。
   4. SNOWNLP：也可以用于情感分词
2. 命名实体标注（NER）
   1. BERT：通过微调，可以实现高效的命名实体识别
   2. SpaCy：具有内置的NER功能，支持多种语言
   3. Flair：一个基于PyTorch的NLP库，支持NER任务
   4. LAC：也支持中文的命名实体识别
3. 关系标注
   1. OpenIE（Open Information Extraction）：用于提取文本中的关系
   2. 关系分类模型（如基于BERT的模型）：可以训练关系分类任务
   3. Spacy
4. 情感标注
   1. TextBlob：简单的情感分析工具，适合英文文本
   2. VADER：专门针对社交媒体文本的情感分析工具
   3. BERT：可以通过微调用于情感分析任务
   4. SNOWNLP
5. 意图标注
   1. Rasa：开源对话系统框架，支持意图识别
   2. BERT：可以通过微调用于意图识别任务
6. 语义标注
   1. BERT和GPT：可用于句子嵌入和语义理解
   2. Sentence-BERT：基于BERT的句子嵌入模型，适合语义相似度计算
7. 事件标注
   1. 事件抽取模型：通常基于深度学习的序列标注模型（如LSTM、BERT 等）进行训练，以识别事件的不同要素
   2. 基于规则的方法：使用模式匹配和模板识别来标注事件

### 综合方案

综合多种考虑，决定使用LAC、SNOWNLP、SPACY和BERT四种模型配合TensorFlow中对应的调整算法实现，其中

1. LAC：实现基本的分词标注、命名实体标注、词性标注
2. SNOWNLP：实现基本的情感标注
3. SPACY和BERT：实现其他的标注
4. TensorFlow：设计人工校正和反馈机制

## 后端语言与架构

### Web服务器的调研与分析

1. **Apache**
   1. 优点：
      1. **成熟稳定**：作为最早的 Web 服务器之一，Apache 经历了多年的开发和优化，稳定性高
      2. **强大的模块化**：支持多种功能模块，如 URL 重写、身份验证、缓存等，灵活扩展
      3. **广泛的社区支持**：拥有庞大的开发社区和文档支持，适合各种规模的网站
      4. **兼容性好**：支持多种操作系统（Linux、Windows、macOS 等）和多种语言（PHP、Perl、Python 等）
   2. 缺点：
      1. **性能问题**：相比 Nginx 和 LiteSpeed，Apache 在处理高并发和静态文件时性能稍逊
      2. **配置复杂性**：高级配置和优化可能需要较为复杂的配置文件，尤其在高流量网站时
2. **Nginx**
   1. 优点：
      1. **高性能**：特别适合处理高并发和静态内容，低内存占用
      2. **反向代理和负载均衡**：天然支持反向代理和负载均衡，常用于高流量网站
      3. **配置简单**：相比 Apache，Nginx 的配置文件更简洁，易于理解
   2. 缺点：
      1. **动态内容处理较弱**：虽然支持 FastCGI 处理动态内容，但不如 Apache 原生支持 PHP 和其他动态内容的处理
      2. **模块不如 Apache 丰富**：没有 Apache 那么多功能模块，扩展性略逊
3. **Tomcat**
   1. 优点：
      1. **专为 Java 应用设计**：如果你主要运行 Java Web 应用（如 JSP 和 Servlets），Tomcat 是一个理想选择
      2. **开源并有广泛支持**：作为 Apache 软件基金会的一部分，Tomcat 是完全开源的，并且有丰富的文档和支持
   2. 缺点：
      1. **主要面向 Java**：对于非 Java 应用，Tomcat 的使用和配置不如其他 Web 服务器方便
      2. **性能较低**：相比 Nginx 和 Apache，在处理静态内容时性能较差
4. **Node.js**
   1. 优点
      1. **高并发**：基于事件驱动和非阻塞 I/O 模型，适合处理大量并发请求
      2. **JavaScript 全栈开发**：前端和后端都能使用 JavaScript，提高开发效率和一致性
      3. **灵活性高**：可以自由选择 HTTP 处理方式和其他功能，适合构建 API 或实时 Web 应用
   2. 缺点：
      1. **不适合传统的静态网站**：作为应用服务器，更适合构建动态内容和实时交互，而不是托管静态文件
      2. **单线程**：虽然非阻塞 I/O 可以提高并发处理能力，但 Node.js 依然是单线程，处理计算密集型任务时可能不如其他服务器

### 最终选择方案

1. **Apache服务器：**
   1. 由于apache服务器的稳定性、成熟性和兼容性能够很好的适配项目需求，故而选择apache作为后端的web服务器
2. **python**
   1. 同时选择的NLP模型和TensorFlow包基本都是使用python实现，python语言也简单，于是选择python作为主要的编程语言
3. **Flask框架**
   1. Flask 是一个轻量级的 Python Web 框架，简洁易学，灵活性高。使用Flask可以轻松集成数据库，网页后端编写等功能

## 前端框架

在后端apache服务器和python编程语言选定后，前端的选择就基本确定：

1. 使用flask自带的Flask-Jinja模版引擎
2. 配合Bootstrap前端框架
3. 利用Bootstrap提供的丰富的 HTML、CSS 和 JavaScript 组件配合Jinja模版设计前端界面

# 系统架构设计

## 系统技术框架

### 后端技术框架

1. 使用linux作为服务器的操作系统
   1. 选择Ubuntu 22.04.5 LTS发行版
   2. 建立在线多用户工作环境
2. 使用apache服务器作为Web服务器
   1. 监听80端口，处理客户端的HTTP请求
   2. 作为反向代理服务器代理flask应用
3. 使用flask框架处理后端逻辑
   1. 处理apache代理转发过来的HTTP请求
   2. 调用LAC，SNOW等NLP模型和TensorFlow处理数据
4. 使用git配合github进行版本管理
5. 使用mysql存储文本数据
   1. 设计良好的数据存储结构
   2. 设计父表存储文件名和文件数据
   3. 设计子表存储每个文本的分词结果
   4. 设计flask调用模块
6. 使用Navicat可视化管理和操作mysql数据库

### 前端技术框架

1. 使用Jinja模版
   1. 方便完成后端向前端的数据传输
   2. 可以使用一些基本的流程控制
2. 使用Bootstrap框架
   1. 利用其丰富的HTML、CSS 和 JavaScript 组件可以搭建简洁美观的前端界面
   2. 可视化展示分词效果和数据特征

## 系统功能框架

### 工作流程

1. 通过公网IP在浏览器上访问文本数据标注网页
2. 手动输入文本或是提交.txt,.csv文件上传数据
3. 点击功能按钮（如分词标注、情感标注），网页回显标注好的数据
4. 给出如显示分词数量、对应分词个数等可视化数据

### 界面设计

1. 界面总体上分为两个部分
2. 左侧位文本数据输入框和文件上传框
3. 右侧上方为功能栏和导航栏
4. 右侧下方为标注数据显示框和数据分析图

# 课程任务分工

|  |  |
| --- | --- |
| 徐文涛 | 负责软件整体设计和后端架构 |
| 罗皓严 | 负责界面设计和前端架构 |
| 李升臻 | 负责NLP模型搭建和算法调试 |
| 杨家豪 | 负责文书工作和PPT设计 |

# 课程进度安排

1. 第6周：集中讲座见面会
2. 第7周：软件课设分组
3. 第8~11周：
   1. 软件需求分析、技术调研及方案论证
   2. 软件初步构建
4. 第12周：中期报告
5. 第13~18周：进一步完善软件的设计与调试
6. 第19周：
   1. 软件演示、验收与提交
   2. 软件源代码，操作说明书
   3. 软件课程设计报告

现在已经基本完成软件的初步构建和基本功能实现（如分词标注情感标注等），在接下来一周中将继续优化网页界面和NLP算法