# A12 企业知识库管理系统

浙江工业大学
ZHEJIANG UNIVERSITY OF TECHNOLOGY

汇报人 / 郑华展

- 艰苦创业 开拓创新 争创一流 -

# 目录

CONTENTS

浙江工业大学
ZHEJIANG UNIVERSITY OF TECHNOLOGY



CONTENTS

- 艰苦创业 开拓创新 争创一流 -

第一部分

# 题目分析

浙江工业大学
ZHEJIANG UNIVERSITY OF TECHNOLOGY

- 艰苦创业 开拓创新 争创一流 -

# 产品定位：“一款优秀的知识库产品”



随着金融行业的日益发展，业务流程越来越复杂，金融产品种类越来越繁多，这就对业务人员工作造成了极大的困扰，如果有一款优秀的知识库产品，可以对相应的知识进行分类管理，快速准确检索，可以极大地减轻业务人员查看相关文档所投入的时间精力，从而可以更好的为客户服务。

**【用户期望】**

（1）搜索准则且效率高，按点击率排序；

（2）页面美观，布局合理；

（3）合理规范的知识审核流程（提供错误自动校验提醒功能）

（4）合理规范的权限控制。

**【任务清单】**

（1）知识的批量上传下载更新；

（2）热点知识统计（点击率，收藏率等）；

（3）知识在线浏览（包含 excel，word，pdf，视频等）；

（4）关键字搜索（包含文本正文）；

（5）完善规范的知识审批流程；

（6）针对知识点建立关联知识，知识评论区等；

（7）知识结构变更支持拖拽等方式。

# 产品定位：“一款优秀的知识库产品”

随着金融行业的日益发展，业务流程越来越复杂，金融产品种类越来越繁多，这就对业务人员工作造成了极大的困扰，如果有一款优秀的知识库产品，可以对相应的知识进行分类管理，快速准确检索，可以极大地减轻业务人员查看相关文档所投入的时间精力，从而可以更好的为客户服务。

能搜索（正文、全拼、首字母。<2s）
能上传下载更新（批量）
能在线浏览
能审核（还要有错误自动校验提醒功能）
能统计热点信息
能评论
能关联知识
能权限管理
能拖拽变更知识结构（前端的活）

第二部分

# 系统功能与框架

浙江工业大学
ZHEJIANG UNIVERSITY OF TECHNOLOGY

- 艰苦创业　开拓创新　争创一流 -

# 知识文档管理板块



模块目标：实现知识资产的统一上传、存储、组织与维护。

模块功能：

- **多格式支持**：Word、Excel、PPT、PDF、TXT、图片（JPG/PNG）、音视频（MP4/MP3）等。
- **批量上传/下载**：支持 ZIP 打包上传与解压入库；支持批量下载选中文档。
- **断点续传**：大文件上传/下载支持断点续传，避免网络中断导致重复传输。
- **在线预览**：集成文档预览服务，无需下载即可查看内容。
- **版本管理**：同一文档支持多版本存储，可查看历史版本、回滚、对比差异。
- **元数据管理**：自动提取或手动填写标题、作者、部门、关键词、创建时间、保密等级等。
- **文件去重**：基于内容哈希（如 SHA256）识别重复文件，避免冗余存储。

# 知识检索与智能搜索板块



模块目标：让用户快速、准确地找到所需知识。

模块功能：

- **全文检索**：基于 ElasticSearch 实现全文索引，支持对文档正文内容搜索。
- **多维搜索条件**：支持按标题、作者、部门、时间范围、文件类型、关键词等组合筛选。
- **拼音首字母搜索**：如输入“qy”可匹配“企业”。
- **模糊匹配 & 同义词扩展**：支持错别字容错（如“知实库”→“知识库”）、近义词联想。
- **高亮显示**：搜索结果中高亮匹配关键词。
- **搜索性能**：响应时间 ≤ 2 秒（百万级文档量下）。
- **搜索历史 & 热门搜索推荐**：记录用户搜索行为，提供智能建议。

# 知识审核与质量控制板块

模块目标：保障入库知识的准确性、合规性与专业性。

模块功能：

- **多级审批流程**：支持自定义审批流（如：提交 → 部门审核 → 知识管理员终审）。
- **自动校验提醒**：
  - 错别字检测（集成语言模型或词典）
  - 敏感词过滤（如涉密、客户隐私等）
  - 格式规范检查（如缺少封面、目录等）
- **审核意见留痕**：审核人可填写意见，提交人可查看并修改。
- **状态标识**：文档状态包括“草稿”“待审”“已发布”“已驳回”“已归档”等。

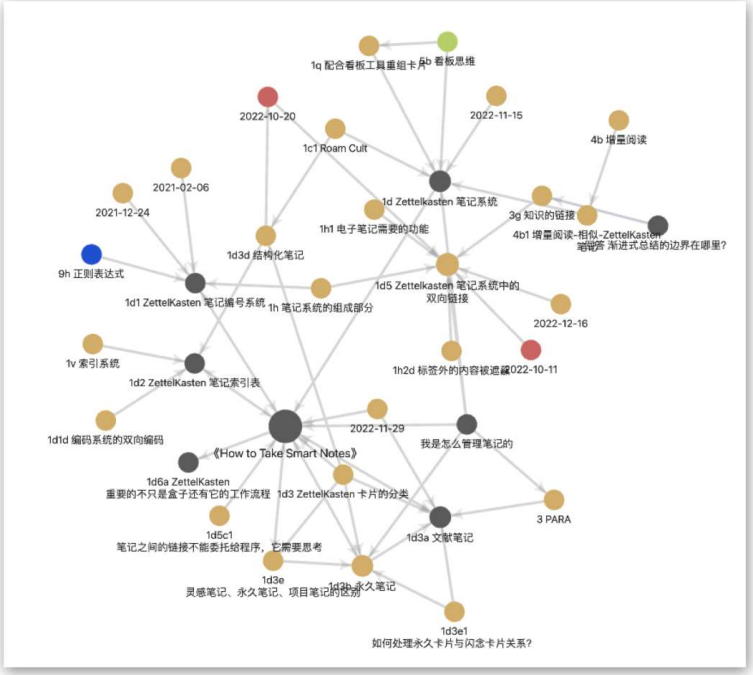
# 知识组织与结构化板块

浙江工业大学 ZHEJIANG UNIVERSITY OF TECHNOLOGY

模块目标：构建清晰、可扩展的知识体系结构。

模块功能：

- **树形知识分类**：支持多级目录（如：行业 → 金融 → 银行 → 核心系统）。
- **拖拽调整结构**：通过可视化界面拖拽节点调整分类层级。
- **知识标签体系**：支持打标签（如“风控”“API设计”“合规”），便于多维关联
- **知识关联图谱**：自动或手动建立文档间的引用、依赖、相似关系。
- **知识地图**：可视化展示知识分布与热点区域。



# 知识互动与运营板块



模块目标：促进知识的使用、反馈与价值挖掘。

模块功能：

- **评论与问答**：每篇文档下可评论、提问，作者或其他专家可回复。
- **收藏与点赞**：用户可收藏常用文档，点赞高价值内容。
- **热点排行**：按周/月统计点击量、下载量、收藏量，生成 Top 10 热门知识榜单。
- **阅读统计**：记录用户阅读时长、完成率（对长文档/视频）。
- **知识贡献排行榜**：激励员工贡献优质内容（按上传量、采纳率等维度）。

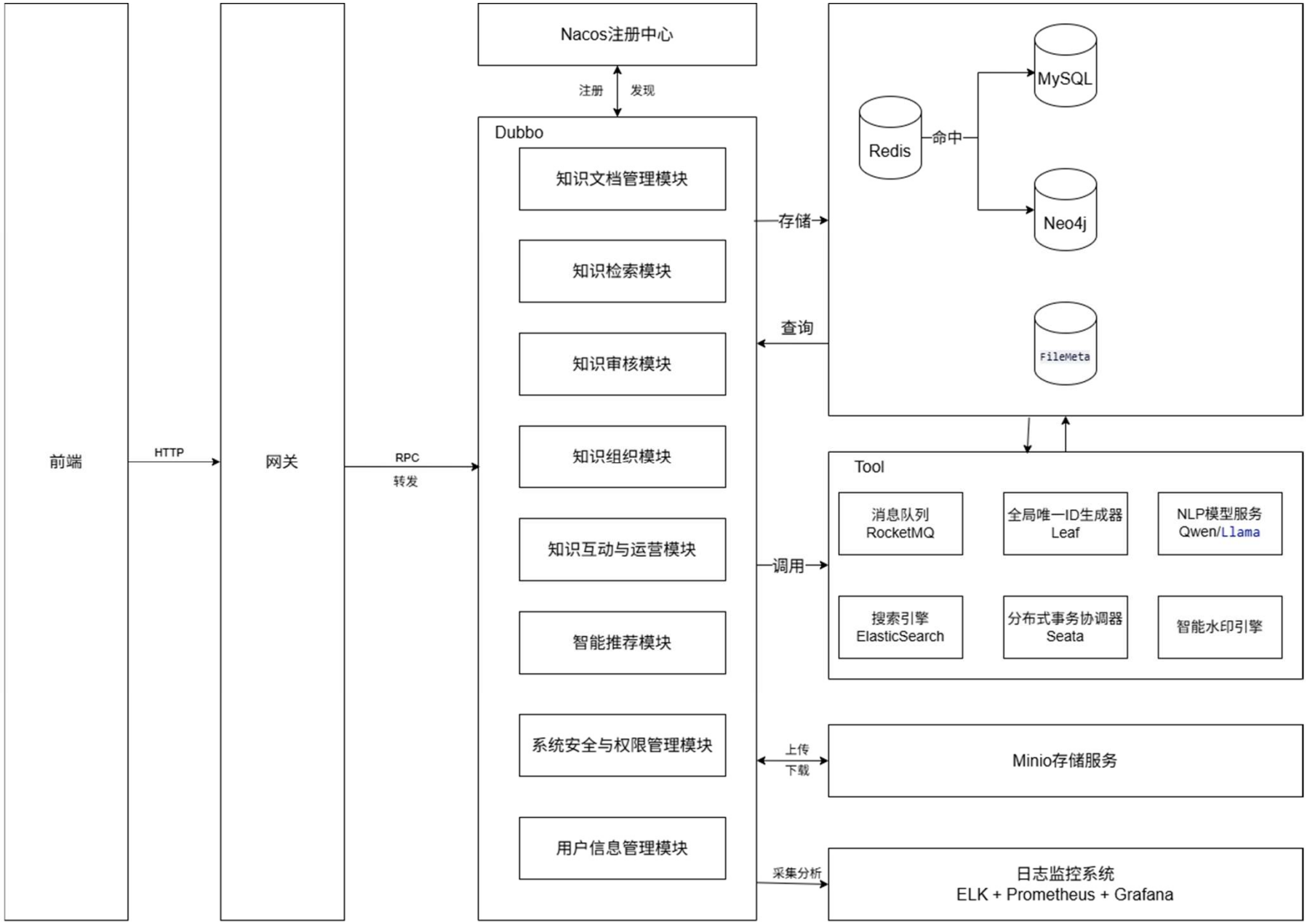
# 系统安全与权限管理板块



模块目标：确保知识资产的安全访问与合规使用。

模块功能：

- **RBAC 权限模型**：基于角色（如普通员工、部门管理员、知识管理员）控制操作权限。
- **细粒度权限**：
  - 查看/下载/编辑/删除/审批 权限可按文档或目录设置
  - 支持“仅预览不可下载”等特殊权限
- **数据加密**：敏感文档支持存储加密（AES）与传输加密（HTTPS）。
- **操作日志审计**：记录所有关键操作（上传、删除、下载、修改权限等），支持追溯。
- **水印功能**：在线预览或下载 PDF 时自动添加用户水印，防止泄露。


前端
HTTP
网关
RPC
转发
Nacos注册中心
注册
发现
Dubbo
知识文档管理模块
知识检索模块
知识审核模块
知识组织模块
知识互动与运营模块
智能推荐模块
系统安全与权限管理模块
用户信息管理模块
存储
查询
调用
上传
下载
采集分析
Redis
命中
MySQL
Neo4j
FileMeta
Tool
消息队列
RocketMQ
全局唯一ID生成器
Leaf
NLP模型服务
Qwen/Llama
搜索引擎
ElasticSearch
分布式事务协调器
Seata
智能水印引擎
Minio存储服务
日志监控系统
ELK + Prometheus + Grafana

第三部分

# 技术选型与方案

浙江工业大学
ZHEJIANG UNIVERSITY OF TECHNOLOGY

- 艰苦创业 开拓创新 争创一流 -

6. 任务要求

【技术路径】

Spring;

struts2;

Ibatis;

apache Lucene;

Nutch。

【技术指标】

搜索响应时间越快越好，要求在 2 秒以内响应。

# Lucene

它是一个**Java 库**。它不是一个完整的应用，而是一个**工具包。也就是说他只是提供了一个API，然后我们还得给他配个分词器**（比如针对中文的首字母和拼音分词器）来实现我们所需的功能

相对底层，开发成本高，相应的，比较灵活

传统的数据库是按**行**存储的，如果你要搜索“分布式”，它得一行一行地扫描。

而Lucene使用的是**倒排索引**。就比如当你搜索“苹果”时，Lucene **直接**查表哪个文档里面有这个词，**索引页**上查“分布式”这个词，索引页直接告诉你它在哪几页出现过，而不是让你从头翻到尾。这个倒排索引的创建过程，就是我们常说的**分词**

需要注意的是：对于中文来说，默认的分词器效果通常很差，这也是为什么我们需要手动配置一下 IK 或 Pinyin 插件

# 想知道怎么分词的？来看看这个。不看算了



如果是英文，他分词相对可以更简单，因为他有空格。直接根据空格分分就差不多了。那中文咋办？

首先明确一点，你建索引的时候怎么分词，你搜索的时候就应该怎么分词，这俩应该是一样的，否则我们就可能匹配不上。

其实他用的是传统的分词算法。如果是IK，那也就是加词典，然后**正向/逆向最大匹配。**

6. 任务要求

【技术路径】

Spring;

struts2;

Ibatis;

apache Lucene;

Nutch。

【技术指标】

搜索响应时间越快越好，要求在 2 秒以内响应。

# Nutch

他是一个**开源的 Web 爬虫框架，**用来抓取网页内容或者建立那种大规模的网页索引（换句话说就是常见的搜索引擎）。

题目里面貌似没提到他需要从网上检索内容?

数据源是**企业内部结构化/非结构化文档**，不是网页

知识库管理系统可以对组织中大量的有价值的方案、策划、成果、经验等知识进行分类存储和管理，积累知识资产避免流失，促进知识的学习、共享、培训、再利用和创新，有效降低组织运营成本，强化其核心竞争力的管理方法。

6. 任务要求

【技术路径】

Spring;

struts2;

Ibatis;

apache Lucene;

Nutch。

【技术指标】

搜索响应时间越快越好，要求在 2 秒以内响应。

貌似还有一个叫做**Meilisearch，没用过因此不说**

# Elasticsearch

它是一个**分布式、可扩展**的**搜索引擎**，**底层使用Lucene**。

**API**是标准的 **RESTful API**，部署和使用相对便利一些。配置一个分词器就可以用了。

使用上，ES有以下几个概念需要了解：

**Index (索引)**：ES 数据的顶级容器，用于划分不同的数据类型。你需要为你的一批文档（比如“用户数据”或“产品数据”）创建一个索引。

**Document (文档)**：ES 中最小的数据单元，一个 JSON 对象。你的一个文档（如一封邮件、一个产品信息）就是一条记录。

**Field (字段)：**JSON 文档中的键值对，比如 title（标题）、content（正文）。

# 数据库的选择



我比较常用的数据库一般就是MySQL和MongoDB，因此我在这俩里面先选选看

MongoDB

首先MongoDB有个缺点，他不支持强事务业务。

而他的好处是可以嵌套文档，无需复杂 JOIN 查询。

但是!

在我们这个项目中，我们很多时候需要一个比较强的一致性（比方说你文档记录和ES里面的记录不能对应错），并且需要多表关联。

貌似…能用上MongoDB的好像就一个评论区，但是这不是系统的主要功能。因此先把他放一边，要用再说

MySQL

这个很常见很常用。

**先看项目本质：这是一个“强关系 + 强事务 + 强权限”的系统**

- 谁（用户/角色）可以看/编辑哪份文档？ → **权限控制**
- 文档从草稿到发布的流程是否合规？ → **审批流**
- 同一份文档修改后要保留历史版本 → **版本管理**
- 财务部的机密文档不能被市场部看到 → **数据隔离**
- 所有操作要可审计 → **日志追踪**

共同点是：**数据之间存在明确关系，且操作必须保证一致性（ACID）**。

因此选MySQL没啥可说的，选就完事了，非常对口。

【任务清单】

（1）知识的批量上传下载更新；

（2）热点知识统计（点击率，收藏率等）；

（3）知识在线浏览（包含 excel，word，pdf，视频等）；

（4）关键字搜索（包含文本正文）；

（5）完善规范的知识审批流程；

（6）针对知识点建立关联知识，知识评论区等；

（7）知识结构变更支持拖拽等方式。

Neo4j他是一个图数据库，以**节点和关系为核心模型**，天然适合表达和计算知识实体间的复杂关联。他提供交互式图谱界面，直观展示企业知识结构。

基于这个玩意我们就可以更方便的实现知识的智能化推荐。但是怎么实现，还得斟酌一下...

主要是得有相关性对吧，那我们需要给他写入每个节点之间的关系。主要有两种方式吧：

**根据关键词建立联系：**编辑上传文档时，手动给他设置关键词或者使用NLP提取关键词，然后把他的关键词写入Neo4j

**基于用户行为建立联系：**也许我们可以这么做，如果很多用户同时读了 A 和 B，则认为 A 和 B 相关。也就是我们可以记录用户行为，然后根据他一段时间内文档同时出现的次数来认为文档之间存在相关性。这当然是一个比较简单的方法，具体还有很多细节需要考虑。

# 分片



顾名思义，分片就是把一个大文件切成多个小块。对于我们这个系统来说分片还是一项比较重要的技术

首先我们来看**上传分片**：

这个是为了实现**断点续传、大文件上传、并发上传。**

前端先要知道文件总大小，然后根据分片大小计算一下分片数量，然后计算出整个文件的哈希和当前传输的分片的哈希。然后把这些信息传递给后端。

| 名称 | 说明 | 示例 |
| --- | --- | --- |
| **File** | 原始文件 | `training_video.mp4` (500MB) |
| **Chunk** | 分片（固定大小） | 每片 5MB，共 100 片 |
| **UploadId** | 本次上传的唯一ID | `u_7a3b8c9d` (UUID) |
| **ChunkIndex** | 分片序号 | `0, 1, 2, ..., 99` |
| **ChunkHash** | 分片内容哈希（可选） | `sha256(chunk)` |

后端接收分片之后，根据之前前端发送的信息校验UploadId是否有效，然后校验ChunkHash是否一致。如果都通过的话就给前端更新上传进度。

确定所有文件都上传了之后，后端得要创建一个最后的文件输出流，按顺序读取每个分片然后写入最终的文件。

# 断点续传



断点续传其实很简单，就是去找你那一次断掉的UploadId对应的上传进程传到哪个分片了，然后通知前端恢复上传就行了。

真假的这么简单？骗你的。我只是一笔带过而已。

“通知前端恢复上传”这个有点问题。你如果是web应用，你前端在浏览器里面运行，然后你就会发现，你找不到原来那个文件路径了。你怎么找都找不到的。比方说一个路径“C:\Users\ZeroHzzzz\jinitaimei.mp4”，你前端实际拿到的路径是：“C:\\fakepath\\ jinitaimei.mp4”

这其实是浏览器的**沙盒机制。**这是为了**给每个网页创建一个受限的执行环境，**保护你的安全。

**那我得咋办?**

**忘记刚刚上传了**整个文件的哈希了吗？作用就是这个。这个也叫做文件指纹。

# 文件去重和文件秒传



文件去重就是为了**避免重复存储相同内容，**文件秒传就是**用户上传已存在的文件时，跳过上传过程，直接返回成功**

所以这俩其实在讲差不多的事情。

前面讲到了分片，我设想中的文件去重其实也是基于分片。

其实就是，我们不是对每个分片取了哈希，然后我们就可以在传输前先把这些分片信息传递给后端，然后后端在库里面找有没有一样的哈希，如果一样说明该分片已经被上传过，因此我们就可以跳过这个分片，后续的话我们还要还原文件（如果需要的话），这个时候给他找到对应分片按顺序还原就行了。这就是**分块去重。**

那这样就会面临一个问题，就是你这个系统由于你得实时预览，因此你得考虑你分片情况下能不能进行预览。你这样一个系统不应该同时存储分片和完整文件，开销太大了。因此我们还是得考虑不同文件不同策略的。

但其实我调研了一下，只有office文档不支持在分片情况下预览，因为他是ZIP封装，他得完整才能解压。图片有不行，其他比如视频、PDF等貌似基本都可以

# 讲不完了要



**其他一些会使用的技术：**

**文件类型识别**（简单的使用魔术字判断一下吧！就是为了避免.exe伪装成.jpg之类的）

**RAG**（检索增强生成，可以实现基于文档内容提问）

**智能水印**（就是不可见水印。在 PDF 元数据或像素中编码用户 ID，即使截图也能追踪）

**文档协作**（目前打算做成版本控制系统。类似Git。同时还得要设置冲突处理机制）

**敏感信息脱敏**（这个更简单，短文本的时候加个正则匹配就行了...）

浙江工业大学

ZHEJIANG UNIVERSITY OF TECHNOLOGY

ZJUT

# 致谢

Thanks

- 艰苦创业 开拓创新 争创一流 -

浙江工业大学
ZHEJIANG UNIVERSITY OF TECHNOLOGY

第一部分 ▶

# 课题背景

SUBJECT BACKGROUND


- 艰苦创业 开拓创新 争创一流 -

第一部分

# 课题背景

SUBJECT
BACKGROUND

- 艰苦创业　开拓创新　争创一流 -

# 在这里输入你的标题