



南京大學
NANJING UNIVERSITY

研究生畢業論文
(申請碩士專業學位)

論文題目	<u>面向民事裁判文書質量 評估平台的设计和实现</u>
作者姓名	<u>黎宇</u>
專業名稱	<u>軟件工程</u>
研究方向	<u>軟件工程</u>
指導教師	<u>劉嘉 副教授</u> <u>何鐵科 助理研究員</u>

2020年5月9日

学 号：MF1832077

论文答辩日期：2020年5月18日

指导教师：何铁科 (签字)



The Design and Implementation of Quality Inspection Platform for Civil Judgement Documents

By

Yu li

Supervised by

Associate Professor **Jia Liu**

Research Assistant **Tieke He**

A Thesis

Submitted to the Software Institute

and the Graduate School

of Nanjing University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Engineering

Software Institute

May 2020

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：面向民事裁判文书质量评估平台的设计和实现

工程硕士（软件工程领域） 专业 2018 级硕士生姓名：黎宇

指导教师（姓名、职称）：刘嘉 副教授 何铁科 助理研究员

摘 要

为了贯彻网络强国战略思想，最高人民法院高度重视信息化建设，先后建成运行中国裁判文书网、中国审判流程信息公开网、中国执行信息公开网、中国庭审公开网四大司法公开平台，实现了司法业务关键节点的信息化及公开化，真正做到便民、透明的阳光执法。作为记录案件完整流程和关键信息的载体，裁判文书是公开信息中必不可少的基础数据。由于数据来源于各个省市的各级人民法院，整理上传过程难免出现纰漏，所以如何衡量数量巨大的司法数据质量，以确保公开数据的真实性和有效性成为技术人员和司法专业人员共同的问题。

本文结合数据质量和信息质量评估领域的知识，将数据根据格式划分为结构化、非结构化两种类型，根据场景的实际需求，从主观及客观两个维度完成评估指标的设计。对于结构化信息，主要为可按关键字抽取得到的标签数据，本文采用数据质量常用的度量方式，结合客观信息论中提到的度量维度对数据质量进行度量。涵盖细致性、延迟性、真实性、完整性、一致性、易读性、准确性共七个类别，结合民事裁判文书场景需求共设计 15 个详细的评价指标。对于数据中的文本类非结构化信息，本系统更关注其中蕴含的语义信息的质量。本文借助有监督的机器学习及深度学习技术，借助系统中积累的大量数据作为带标签的信息质量高的训练正样本，模拟人为的信息质量评估过程对裁判文书进行评估。

借助以上指标的设计和计算方法，本文基于 Flask 框架完成了民事文书质量度量平台的设计和实现。后端存储使用 SQLAlchemy 完成数据库表及数据到代码中类与对象的映射，借助 Flasgger 工具编写接口文档并完成测试，为前端提供符合 Restful 规范的接口，同时借助 Redis 作为消息存储中间件，使用 Celery 实现分布式异步任务调度。平台包括数据交互，用户数据管理，质量检测，统计分析四大模块，为用户提供数据上传，文书解析，质量评估，评估结果可视化的全流程质量度量服务。

关键词：民事裁判文书，客观信息质量，数据质量，自然语言处理

南京大学研究生毕业论文英文摘要首页用纸

THESIS: The Design and Implementation of Quality Inspection Platform
for Civil Judgement Documents

SPECIALIZATION: Software Engineering

POSTGRADUATE: Yu li

MENTOR: Associate Professor Jia Liu, Research Assistant Tieke He

Abstract

In order to implement the strategic thought of building China into a cyberpower, the Supreme People's court attaches great importance to information construction. They have successively built and operated four major judicial open platforms, including China Judgements Online, China Judicial Process Information Online, China Execution Information Online, and China Court Trial Online. It realized The informatization and openness of key nodes of judicial business, which truly facilitate the public and transparent law enforcement. As the carrier of recording the whole process and key information of the case, the judgment document is the essential basic data in the public information. As the data comes from the people's courts at all levels in various provinces and cities, it is inevitable to make mistakes in the process of sorting and uploading. Therefore, how to measure the quality of judicial data with a huge amount to ensure the authenticity and effectiveness of the open data has become a common problem for technicians and judicial professionals.

Based on the knowledge of data quality and information quality assessment, this paper divides the data into structured and unstructured according to the format. Based on the actual needs of the scene, the design of assessment indicators is completed from subjective and objective dimensions. For structured information, mainly including tag data which can be extracted by keywords. In this paper, data quality is measured by the commonly used measurement method, combined with the measurement dimension mentioned in the objective information theory. It covers seven categories: delicacy, delay, authenticity, integrity, consistency, readability and accuracy. 15 detailed evaluation indexes are designed according to the requirements of civil judgment documents.

For the unstructured information of text type in data, we pay more attention to the quality of semantic and pragmatic information contained in it. In this paper, with the help of supervised machine learning and deep learning technology, we take the large number of data accumulated in the system as positive training samples with high information quality to simulate the artificial information quality evaluation process.

With the help of the index design and calculation method above, this paper uses the Flask framework to complete the design and implementation of the civil document quality measurement platform, and use the SQLAlchemy in the backend storage to complete the mapping of database tables and data to classes and object in code. Besides, we also use the Flasgger to generate the interface document and test automatically, providing the interface which obey the Restful standard. In order to implement distributed task scheduling, we use Celery with Redis as the message broker to distribute the asynchronous tasks. The platform includes four modules: user management, data interaction, quality inspection and model training. It provides users with the whole process quality measurement services of data upload, document analysis, quality evaluation and evaluation results visualization.

Keywords: civil judgement document, objective information quality, data quality, natural language processing

目录

表 目 录	x
图 目 录	xii
第一章 引言	1
1.1 项目背景	1
1.2 国内外研究现状	2
1.3 研究目标与研究内容	3
1.3.1 民事裁判文书度量体系的构建	3
1.3.2 文书评估平台的设计和实现	4
1.4 本文组织与结构	5
第二章 技术综述	7
2.1 裁判文书质量评估	7
2.1.1 数据质量相关研究综述	7
2.1.2 信息质量相关研究综述	8
2.2 自然语言处理技术	9
2.2.1 非平行式文本风格迁移技术	9
2.2.2 BERT 技术综述	10
2.3 系统使用开发技术	11
2.3.1 web 应用框架 Flask	11
2.3.2 分布式任务队列 Celery	12
2.3.3 对象关系映射工具 SQLAlchemy	13
2.3.4 Restful 接口测试工具 Flasgger	13
2.4 本章小结	14
第三章 民事裁判文书质量评估指标设计	15
3.1 客观质量评估指标设计	15

3.1.1	客观信息质量度量指标设计	16
3.1.1.1	细致性	16
3.1.1.2	延迟性	16
3.1.1.3	真实性	17
3.1.2	客观数据质量度量指标设计	18
3.1.2.1	完整性	18
3.1.2.2	一致性	19
3.1.2.3	易读性	20
3.1.2.4	准确性	22
3.2	主观质量评估指标设计	24
3.2.1	引用法条合理性度量	24
3.2.2	语言风格度量	25
3.2.2.1	语言风格鲜明性	25
3.2.2.2	语言情感客观性	25
3.2.2.3	文本重构性检测	25
3.3	本章小结	26
第四章	民事裁判文书质量评估系统需求分析及设计	27
4.1	系统边界	27
4.2	系统需求分析	28
4.2.1	用户数据交互及管理	28
4.2.2	文书质量检测与统计分析	32
4.2.3	系统功能性需求	34
4.2.4	系统非功能性需求	34
4.3	系统概要设计	35
4.3.1	系统整体框架	35
4.3.2	4+1 视图	37
4.4	数据交互模块设计	42
4.5	用户数据管理模块设计	43
4.6	质量检测模块设计	45
4.7	统计分析模块设计	47

4.8	数据库设计	48
4.8.1	ER 图	48
4.8.2	数据库表设计	49
4.9	本章小结	51
第五章	民事裁判文书质量评估系统实现	53
5.1	数据交互模块实现	53
5.2	用户数据管理模块实现	54
5.3	文书质量检测模块实现	56
5.3.1	客观度量指标的实现	56
5.3.2	主观度量指标的实现	60
5.4	统计分析模块实现	70
5.5	本章小结	71
第六章	民事裁判文书评估系统测试与分析	73
6.1	测试目的	73
6.2	测试环境	73
6.3	系统功能测试	73
6.4	系统性能测试	77
6.5	测试结果	77
6.6	本章小结	79
第七章	总结与展望	81
7.1	总结	81
7.2	未来工作与展望	81
	参考文献	83
	简历与科研成果	87
	致谢	89
	版权与原创性说明	91

表 目 录

1.1	文书语义质量特征度量表	3
2.1	数据质量常见度量维度表	8
3.1	客观信息质量度量表	15
3.2	客观数据质量度量表	15
3.3	民事裁判文书书写过程中的标点符号规范表	20
3.4	民事裁判文书中构成事项及说明表	22
3.5	民事裁判文书中对参诉人信息记录的要求说明表	23
4.1	用户数据上传用例描述表	29
4.2	质量检测结果下载用例描述表	30
4.3	用户配置管理用例描述表	30
4.4	用户文书管理用例描述表	31
4.5	用户任务管理用例描述表	31
4.6	文书预处理用例描述表	32
4.7	文书质量检测用例描述表	33
4.8	质检结果统计分析用例描述表	33
4.9	功能需求分析表	34
4.10	<i>judicial_docs</i> 表	49
4.11	<i>reports</i> 表	50
4.12	<i>users</i> 表	50
4.13	<i>tasks</i> 表	50
4.14	<i>configs</i> 表	50
4.15	<i>analysis_reports</i> 表	51
5.1	度量指标及文书篇章结构对应关系表	56
5.2	客观评价指标度量结果表	62
5.3	知网中文程度级别词语分类说明表	67

5.4	主观评价指标度量结果表	68
6.1	系统测试环境说明表	73
6.2	用户数据上传测试用例描述表	74
6.3	用户数据下载测试用例描述表	74
6.4	用户配置管理测试用例描述表	75
6.5	用户文书管理测试用例描述表	75
6.6	用户任务管理测试用例描述表	76
6.7	文书质量检测测试用例描述表	76
6.8	质检结果统计分析功能测试用例描述表	77
6.9	系统性能效率测试用例描述表	77
6.10	系统测试用例执行结果表	78
6.11	人工验证实验结果表	79

图 目 录

2.1	信息质量度量评估框架图 [1].....	9
2.2	非平行式文本风格迁移模型结构图 [2]	10
2.3	Celery 基本工作流程图.....	12
3.1	诉讼费承担说明完整性流程图.....	19
3.2	争议焦点检测流程图	21
3.3	参诉人信息准确性流程图	24
4.1	系统边界图	27
4.2	用户数据交互及管理模块用例图	29
4.3	文书质量检测与统计分析模块用例图	32
4.4	系统整体架构设计图	36
4.5	文书质量评估系统逻辑视图.....	38
4.6	文书质量评估系统开发视图.....	39
4.7	文书质量评估系统进程视图.....	40
4.8	文书质量评估系统物理视图.....	41
4.9	数据交互模块设计类图	43
4.10	用户数据管理模块设计类图.....	44
4.11	文书质量检测模块设计类图.....	45
4.12	统计分析模块设计类图	47
4.13	ER 图	48
5.1	数据交互模块顺序图	53
5.2	系统文书上传页面	54
5.3	系统文书管理页面	54
5.4	系统任务管理页面	55
5.5	系统配置管理页面	55
5.6	文书数据集地域分布频数图.....	57

5.7	文书数据集时间分布频数图	57
5.8	细致性度量值数据统计 violin 图	58
5.9	细致性度量值数据统计 violin 图	58
5.10	案件信息延迟性统计分布图	59
5.11	裁判文书延迟性统计分布图	59
5.12	文书关键段落文本长度统计 violin 图	60
5.13	文书关键段落文本长度统计密度图	61
5.14	LDA 模型训练关键代码图	63
5.15	文书文本数据样例图	64
5.16	司法类新闻文本数据样例图	64
5.17	基于 CNN 的文本分类模型结构图	65
5.18	CNN 训练过程 loss 变化图	65
5.19	基于情感极性词典的情感分析流程图	66
5.20	文书质量检测模块顺序图	68
5.21	系统单篇文书质量报告页面	69
5.22	统计分析模块顺序图	70
5.23	系统批量检测统计分析报告页面	71
6.1	系统度量耗时统计分析	78

第一章 引言

1.1 项目背景

为推进“数字法制，智慧司法”建设，实现司法领域的全业务线上办理，全流程依法公开，人民法院先后建成中国裁判文书网，中国审判流程信息公开网，中国执行信息公开网即中国庭审公开网四大平台，完成了审判过程、庭审、文书、执行四个重要节点的全透明，实现了司法建设改革的第一步，也是智慧司法建设的基础。

裁判文书网作为最早建成的公开平台，从2013年7月1日上线起积累了大量的裁判文书数据，目前各类文书总量已达8800余万篇。作为人民法院审判活动的静态体现，裁判文书的公开不仅为法官判决提供了数据支持，也成为人民群众监督审判公正性的窗口，文书网的累计访问频次高达400亿，并且随着最高人民法院英文网站暨新版中国裁判文书网的上线，越来越多境外的用户也加入了文书网的用户行列中，面对来自全球数以亿计的用户，公开文书的质量好坏就显得尤为重要。

最高院指出，裁判文书作为人民法院执行审判权的载体，其质量是司法能力的集中体现，而由于裁判文书来自不同地域、不同级别的人民法院，撰写者的法律素养和文字功力也不尽相同，导致裁判文书的质量参差不齐，甚至存在错别字等低级错误。为了提高文书质量，各级法院要求公开的裁判文书上网前必须经过层层审批，确保无误后方可公开，而随着案件数量的增加，这种冗余繁琐的人工检查过程无疑是对司法资源的浪费，并且也极大地影响了裁判文书公开的及时性。

本论文使用自动化度量的方法，从主观客观两个角度对裁判文书完成度量，不仅可以识别格式不规范等语法错误，也可从语义角度衡量文书的内容质量，建立完整的文书质量度量体系。调查显示公开的裁判文书中，民事案件的裁判文书数量占六成以上，是审判案件中的重要组成部分，并且由于涉及案件关乎民生，用户检索浏览的次数也较多，所以本论文选取民事一审裁判文书作为数据基础，根据最高院的样式规范和审查过程中出现的问题定义度量指标，实现对民事一审裁判文书的质量度量。

1.2 国内外研究现状

裁判文书在我国司法活动中扮演着极为重要的角色。裁判文书不仅记载了诉讼活动关键节点信息，也是人民法院确定和分配实体权利义务的凭证，以及上级法院监督下级人民法院审判公正性的重要凭据。高质量的裁判文书可以为法律共同体提供判例研究的素材，保证法律审判的一致性和公平性。除此之外，随着“智慧法院”建设的推进，如何实现网络公开的海量裁判文书质量的自动化质量度量成为难点。由于各个国家法系不同，裁判文书的书写规范和内容也不尽相同，所以大部分针对我国法系的裁判文书的研究多为国内学者所著。

在裁判文书公开平台建设前，裁判文书的质量研究主要集中在内容规范，研究人员也以司法专业或语言专业人员为主。胡云腾在 [3] 中提到裁判文书说理的重要性，认为说理是司法民主和理性相结合的体现，并指出我国大部分裁判文书存在千篇一律、说理不当甚至不愿说理的缺陷。张慧清在 [4] 中对英美法系裁判文书制作中体现的论证性，选择性的特点做了详细分析，并根据大陆法系和英美法系制作进程的差异，结合我国实情提出了对裁判文书改革的方向，其中不但包括语言规范，繁简得当等语法规规范要求，也提到使用法律合理性和认定事实描述清晰的重要性。[5] 结合判例样本，以三段论为基础构建了裁判文书通用说理模式，对以后的裁判文书的释理过程的书写起到指导参考的作用。

早期对于裁判文书质量的研究大部分集中在理论阶段，意在裁判文书的书写起到指导作用，少有研究者提出对已有裁判文书的质量度量方法。[6] 中提出了裁判文书说理的四大静态指标，分别为语言文字、事实部分、裁判理由以及裁判结果，对于每个指标又详细制定了若干二级指标，并且对每个指标设定了固定的权重。杨贝也在其论文中提到将理论转化为评价机制量化裁判文书质量，[7]，研究以 2017 年北京市判决书的论证质量调查为例，将三段论和司法专业相结合，提出裁判文书论证质量的度量指标并论证了其可行性和可靠性。两种裁判文书的度量指标都建立在专家知识和实践认证的坚实基础上，但二者都是基于人工评分，实际操作过程中难免存在主观认知导致的偏差，除此之外，日益增加的裁判文书数量和冗余的质检工作极大的增加了司法人员的工作量。

随着计算机技术的发展，如何借助信息化时代的成果度量海量的裁判文书质量成为计算机领域的研究者们需要解决的问题。目前已有的与裁判文书相关的交叉学科研究以裁判文书的纠错系统为主。[8] 根据汉语言文学和裁判文书的特点，针对易错的词语、句子、标点、结构、法律条文引用和司法领域专用术语使用六个领域，使用 n-gram 模型建立了可信的二元搭配数据库，作为司法文书智能校对系统的基础。现有的裁判文书纠错系统也多使用构建好的基准库对原始文书进行匹配，将检测结果以报告或高亮的形式展现给用户，不仅可作为质

量评估的结果，也可辅助司法人员对未完成的裁判文书进行修改，从根本上提升裁判文书的质量。

除了针对规范化后的裁判文书进行语法质量度量的研究，也有一些研究考虑到深层语义质量的探索。李嘉伟在论文中 [9] 提出对裁判文书说理全面性和引用法条准确性的度量，作者分别使用 XG-Boost 以及深度残差网络完成了问题的建模，并在人工标注的数据集上进行了充分的实验，验证了使用深度学习的方法量化裁判文书语义质量的可行性。廉昊 [10] 则结合客观信息论和粗糙集领域知识，结合裁判文书固有的特点，制定了裁判文书内容和语义度量指标，具体指标项及计算方式如表1.1所示，

表 1.1: 文书语义质量特征度量表

语义特征	度量方法描述
谓词句法树角色	语义谓词在语法书中角色
谓词词性	谓词在句法树中词性
谓词数目	语义谓词本身的数量特征
语义角色	语义角色占比
语义角色范围内句法树角色	语义角色范围内句法树角色比例
语义角色范围内的词性	语义角色范围内单词的句法树角色比例
语义角色类内句法树角色	语义角色类内句法角色占比
语义角色类内的词性	语义角色范围内单词的词性角色占比

1.3 研究目标与研究内容

本论文首先完成文书质量度量体系的构建，定义客观及主观两个角度的度量指标及计算方法，实现多维度全方位的裁判文书质量度量，然后在此基础上完成文书质量检测平台的设计和实现，达到自动化文书质量检测的目的，同时将度量结果可视化的显示给用户，增强评估结果的可解释性和可信度。

1.3.1 民事裁判文书度量体系的构建

论文主要从主观及客观两个角度完成文书质量的评估，客观指标为文书规范性表层质量度量，主观指标为文书深层信息质量度量。

1. 客观质量度量体系构建

裁判文书具有司法领域文本特有的规范性和标准性，所以文书数据可通过第三方解析软件解析为 xml 格式。本文引用信息质量及数据质量度量的方法，抽象出最常用的细致性、延迟性、真实性、完整性、一致性、易读性、准确性七个评估指标，结合最高院颁布的民事裁判文书的书写规范，共制定了 15 种民事裁

判文书适用的详细度量方法，通过抽取 xml 文书中的标签信息和文本信息，完成对文书规范性和客观信息质量的评估。

2. 主观质量度量体系构建

文书除了表层客观性质量，还需考察其内容的语义质量，而这种质量的评估多取决于读者的主观判断。本文拟借助自然语言处理中的方法，以优秀裁判文书数据为标准数据完成模型预训练，使用训练好的模型对待测数据进行评估，模拟人工打分过程。评估指标主要包括引用法条合理、语言风格鲜明性、情感客观性和语言重构检测四部分。对应的研究方法分别为，通过基于 CNN（卷积神经网络）分类算法评估司法文本语言风格鲜明性，利用基于 BERT 的文本情感分析算法度量文书中描述性文本的客观性，使用文本间编辑距离及序列相似性衡量文书关键段落之间的重构关系，以及借助 LDA（隐含迪利克雷分布）主题模型挖掘文书隐层的关联关系，进而衡量文书引用法条的合理性。最后通过对上述评估指标的有机融合，得到文书主观质量评估的结果。

1.3.2 文书评估平台的设计和实现

基于上述的文书质量检测体系，本文实现了文书质量评估平台的后端计算和存储功能，为前端可视化页面提供数据接口，主要包含以下四个模块：

1. 数据交互模块

支持用户上传单篇或多篇裁判文书，平台会将其存储到后端服务器以使用户后续的调用。对于生成的质量检测结果和分析结果，用户可以选择下载前端可视化报告页面的 pdf 版本，同时系统也为用户提供 json 格式的详细检测结果下载，以使用户进行自定义的统计分析操作。

2. 个人数据管理模块

系统为用户上传的文书提供管理操作，为了方便用户进行批量文书检测，系统支持用户一次性选择多篇文书创建检测任务，并可对已创建的任务进行增删改查。除此之外，为了满足用户自定义质量检测指标的需求，系统允许用户更改指标配置，并在用户提交质检任务时默认使用用户自定义的配置文件进行质量检测。

3. 质量检测模块

质量检测模块和系统的核心业务模块，为用户提供包含细致性、延迟性、真实性、完整性、一致性、易读性及准确性共计七个类别的客观检测指标，详细指标共计 15 项。以及语言风格鲜明性、情感倾向性、语言重构检测、引用法条合理性的四个主观指标检测服务，系统将根据用户配置对指标进行组合后完成文书质检，并将质检结果以 json 格式发送给前端进行可视化页面展示，同时以文

件形式存储在后端服务器方便用户重复调用或下载。

4. 统计分析模块

对于批量文书检测任务，系统在对任务内文书进行单篇质检的基础上，会针对任务内文书的质检结果进行统计分析，方便用户了解文书集的整体数据质量，对于统计结果，系统同样支持用户下载可视化 pdf 报告或 json 数据格式报告两种形式，同时支持用户批量下载数据集内包含的单篇文书质检结果。

1.4 本文组织与结构

论文的组织结构及各章简介如下：

第一章为引言部分，简要介绍项目的研究背景，以及该研究领域在国内外的研究概况，然后介绍本文的主要研究目标与研究内容。

第二章为本文中用到到的相关技术领域的综述。主要包括两方面，一方面为客观度量指标设计中引用的数据质量和信息质量相关研究，和主观度量中使用到的自然语言处理技术介绍，另一方面为文书质量评估平台实现中涉及的相关开发技术介绍。

第三章为文书质量评估指标的设计。本章详细介绍了裁判文书主观质量度量与客观质量度量用到的指标及计算方法，其中客观质量包括细致性、延迟性、真实性、完整性、一致性易读性、准确性七类 15 个详细度量项，负责度量文书表层信息规范性；主观指标负责挖掘文本深层信息质量，从语言风格鲜明性，文本重构性，文本情感客观性以及引用法条合理性四个角度完成度量。

第四章为裁判文书质量检测平台的需求分析和设计。本章首先对平台的用户功能需求做了详细分析，然后对平台整体框架及实现逻辑进行概要介绍，最后将系统划分为数据交互、用户数据管理、文书质量度量项以及统计分析四个模块，针对每个模块展开详细的分析与设计。

第五章为系统实现。本章按照第三章中系统设计的模块划分，分别展示了系统各个功能模块的顺序图和核心代码，以及相关的前端展示页面。

第六章为系统测试与分析。本章从系统功能性及性能两个角度出发，以设计测试用例的方法测试系统是否满足第三章提出的需求及设计部分需求。测试结果表明评估平台具有一定的实用性和可靠性。

第七章总结与展望。总结全文，并提出研究存在的问题和未来的改进方向。

第二章 技术综述

2.1 裁判文书质量评估

2.1.1 数据质量相关研究综述

20 世纪 90 年代，随着信息科技的快速发展，外国学者开始数据质量的研究，并从不同的角度对数据质量做了定义，由 MIT 的 Richard Y 带领的数据质量管理团队将数据质量定义为“数据满足使用的程度” [11]，并且提到数据质量的评估取决于数据的使用者，同时，他们将“数据质量维度”定义为“一组数据质量属性”，这些属性反映了数据在不同类别上的质量好坏，该理论也奠定了数据质量评估的基础框架。

早期提出的指标大多基于理论或借鉴其他学科已有成果，例如 [12] 基于理论提出的准确性、完整性、最小性和一致性等，但并没有定义详细的评估方法。为了统一各个应用领域的质量维度定义，[13] 参考本体论提出的质量维度，或者 [14] 借用数据工程领域的衡量指标，将指标划分为主观的数据质量参数和客观的数据质量指示器。后来随着信息技术的发展，数据质量评估大多基于网络数据。例如 Alexander 和 Tate 在 1999 年提出的六个评估标准，其中包括权威性、准确性、客观性、通用性、覆盖范围程度以及可交易性 [15]，Gauch 在 2000 年针对信息检索系统中的数据质量提出的六个维度，包括流通性，可用性、信息噪声比、权威性、流行程度以及聚集程度 [16]。可以看出不同的指标评价体系可能包含重复的评价指标，在前人研究的基础上，Knight 和 Burn[17] 总结了最常见的维度，以及它们被纳入不同数据质量评估框架的频率，其中使用频率最高的 5 个指标如表 2.1 所示。

相比较而言我国对数据质量的研究起步较晚。2008 年，中国人民解放军总参谋部第 63 研究所成立了数据质量研究小组。他们提出了数据质量研究中的一些基本问题，例如数据质量的定义、错误源、以及质量管理中的改进方法等 [18]。2009 年，中科院计算机网络信息中心提出了数据质量评价方法和指标体系，将数据质量分为外部形式质量、内容质量、内容质量三大类，以及质量的效用，每一类又细分为质量特性和评价指标。2011 年，西安交通大学成立了信息质量研究小组，从过程、技术、管理等多个角度分析了大数据质量时代，我们需要面临的挑战及应对措施 [19] 之后，[20] 在其基础上从数据用户的角度，提出了分层次的质量框架，其中包括大数据质量维度、质量特征和质量指标，并在此框架的

表 2.1: 数据质量常见度量维度表

度量维度	引用次数	维度定义说明
准确性	8	定义为数据正确、可靠并且无误
一致性	7	定义为数据以相同格式呈现并与以前数据兼容
安全性	7	定义为为维护信息的安全对信息的访问控制设置
及时性	7	定义为对于当前任务, 信息是否足够新
完整性	5	定义为数据不会丢失, 并且对当前任务具有足够的广度和深度
简洁性	5	定义为数据在表述的过程中言简意赅

基础上构建了动态评估过程。

2.1.2 信息质量相关研究综述

信息质量的研究多起源于公司或机构由于信息质量差导致的重大事故。事故不但涉及到金融机构上亿美元的错误订单 [21], 或者电信公司百万美元的票据损失等巨额的经济损失 [22], 还会造成“文森特”号误射伊朗客机, “挑战者”号爆炸等人员伤亡事件 [23], 严重时可能会影响国际之间的军事或政治关系。由此可见信息质量研究的重要性。很多机构都在经历了损失后逐渐重视信息质量的研究, 2008 年全球金融危机暴露了美国金融体系的内在缺陷, 危机后花旗成立专门部门 CDO(Chief of Data Office 数据办公室) 负责提高信息质量 [24][25]。

信息质量研究分为两种途径, 分别为管理派和数据库派, 数据库派是指在已有数据的基础上, 对信息的质量进行的研究, 管理则是从日常的业务方面入手, 通过分析用户和生产者的信息交互过程和结果数据, 完成信息质量的评估。鉴于本论文中的信息对象为已有的裁判文书, 所以主要调研方向集中在数据库派。对于已有数据, 信息质量的评估与数据质量相似, 都是通过分别衡量其在设定好的各个维度上的得分或类别, 最后汇总得到整体的质量评估结果。

根据 [1], 可将其划分为三个层次, 如图 2.1 所示, 分别为度量层、维度层和方法层。其中度量层代表用户视角和数据视角, 分别在背景依赖和背景独立两种情况下度量数据; 维度层包含各种确切的数据维度指标, 和上一层的维度层相对应, 其中维度和指标为多对多的对应关系; 再上层为方法层, 主要涉及信息质量评估的框架和模型等。信息度量的方法和框架可以适用于多种信息度量场景, 但其中的度量维度却需根据需求定制化设计, 同时还要尽可能多的覆盖领域内数据, 保证其普适性。

很多研究者指出, 信息质量与数据质量不同。[26] 中对数据和信息的关系给

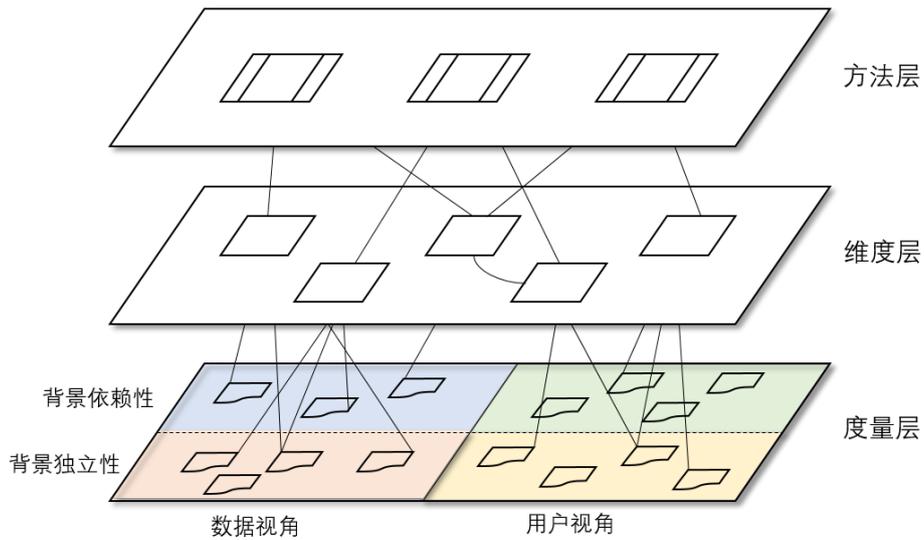


图 2.1: 信息质量度量评估框架图 [1]

出“信息是数据和背景的加和”的定义，背景代表预先设定的领域内符号的含义，数据的使用者只有在具有了背景知识作为基础的情况下，才能获取数据中包含的信息，换句话说，数据是信息的载体，信息是数据传递的内容。张博指出高质量的数据不代表其中包含的信息也同样质量高 [27]。但 Kelin 也提出，数据质量和信息质量会根据研究者自己的观点呈现出不同的特征 [28]。对于本论文中研究的对象裁判文书，我们无法界定其质量属于数据质量的范畴还是信息质量的范畴，所以两种质量评估的方法和维度对我们的研究都具有参考价值，我们不仅要从数据本身的角度衡量，也要从用户的角度考虑评估其内容质量对用户的价值，丰富评价指标。

2.2 自然语言处理技术

2.2.1 非平行式文本风格迁移技术

非平行式迁移在计算机视觉领域有广泛的研究基础，Gatys 曾经提出一种方法 [29]，通过抽取两张图像的内容和样式特征，然后将一张图像的内容和另一张图像的样式结合起来，即可获得风格迁移后的图像。在非平行式的学习中，最核心的问题在于如何对齐两个数据域，研究者们提出了不同的解决方案，例如 CoupledGANs[30] 采用了网络之间共享权重的方法来学习数据的跨域表示，CycleGAN[31] 通过引入循环一致性来缓解传递性，达到使传递函数正则化的目的。这些方法都在图像领域的风格迁移任务中取得了很好的表现，但是由于自

然语言具有离散型，所以上述方法并不能够完整迁移到自然语言领域，研究者需要重新开发新的方法完成文本迁移任务。

在自然语言处理的研究领域，大多数涉及生成的任务都需要使用平行标注的训练数据完成训练，例如翻译和摘要生成任务。如果没有平行标注的数据集，则需要从简介训练标记中引导句子生成。Muller 等人提出通过控制文本的隐藏表示来生成语句 [32]，以达到满足分类器对指定属性的分类需求，例如文本的情感属性。Hu 等人通过学习分离的潜在表征生成属性可控的句子，他们的模型建立在变分自动编码器的基础上，并使用独立约束来确保属性能够被从生成的句子中推断出来。

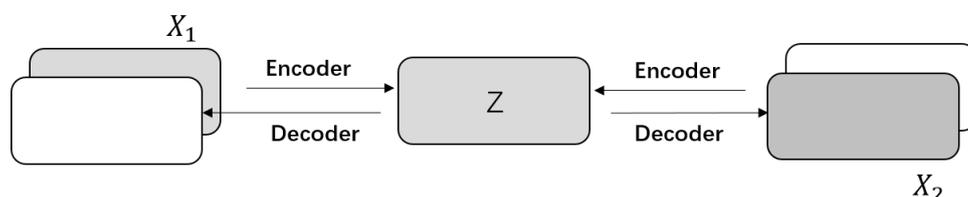


图 2.2: 非平行式文本风格迁移模型结构图 [2]

本文中使用的的方法 [2] 即建立在分布式交叉对齐的基础上，以实现样式转换和内容保存。具体的方法为以一个句子及其原始的风格表示作为输入，学习能够将其映射到独立风格表示的编码器。然后再将其传递给风格相关的解码器进行渲染。具体算法模型如图2.2所示， X_1 和 X_2 表示具有 y_1 及 y_2 两种风格的文本域， Z 是两个文本域共享的潜在内容域，编码器 *Encoder* 将句子映射为内容表达，然后再由 *Decoder* 生成器根据风格生成新的句子加入对应文本域中。模型从文本风格迁移生成的句子中获取额外信息，从而得到两个分布式对齐约束。例如积极情感的语句通过文本风格之后得到的消极情感语句，应该和给定的消极情感语句对齐。

2.2.2 BERT 技术综述

预训练词向量在很多自然语言处理任务中扮演着重要的角色，尤其对于数据量较少，包含信息不足的学习任务。一个好的词向量不但可以反映出语法和语义的复杂特征，更能根据不同的上下文信息进行变换。词向量是为了将人类的自然语言转换成计算机可理解的符号而设计的表达方式，早期最简单的向量表示为 one-hot 模式，每个单词都用长度为单词表长的向量表示，在该单词的序列位置值为 1，其余都为 0。这种向量表示维度较高，且无法表示词语之间的关系。word2vec[33] 克服了上述两个缺点，他的维度通常在 100-300 之间，训练过程主要分为两种形式，分别为根据单个词语预测上下文的 skip-gram 模型，以及

根据上下文预测单个词的 CBOW 模型，这两种词向量的训练都可以在无监督模式下完成，充分利用网络积累的海量文本数据。

基于 word2vec 训练的词向量在很多任务中取得了很好的表现 [34]。但它也无法区分近义词和反义词，并且由于单词和向量是一一对应的关系，也无法捕捉到不同词语在不同上下文中的关系，所以不能解决一词多义的问题。研究者们引入了动态向量的概念，不再使用固定的映射表来定义词向量。Peters 在论文中提出了 ELMo 模型。[35]，和传统的前向 LSTM 语言模型相比，ELMo 不但增加了 LSTM 的层数，并且加入了后向语言模型，对应的输出也不再是固定的词向量，而是一个多层的 BiLSTM 模型，实际使用的过程中需要将句子输入到模型中，得到每个时间步在每一层的输出，最后在具体的 NLP 任务中线性加权得到最后的词汇表示。这种方法很好的解决了多义词的问题，并且在 SQuAD,NER 等多个任务上取得了显著提升。但由于其使用单向语言模型的拼接，在特征融合方面难免会有欠缺。

BERT 模型汲取了上述方法提到的优点并解决了存在的问题 [36]。BERT 也采用动态向量的模式，并且相比于之前的单向语言模型的拼接，BERT 使用了遮蔽语言模型 (masked language model,MLM) 来实现预训练的双向深度表示，也被称作 cloze 任务 [37]，该模型允许使用被遮挡词上下文的特征，从而得到一个深度双向的 Transformer。除此之外，BERT 中加入了句子级别的模型关系，通过二进制的下一句预测任务完成训练，捕获了句子级别的上下文关系，对很多 NLP 任务例如问答、自然语言推理等都有很大帮助。虽然 BERT 的训练过程耗时耗力，但很多下游任务都可以在开源的模型上进行微调，本文中的实验使用了谷歌开源的中文模型作为基础，使用预处理好的数据作为微调的训练数据。

2.3 系统使用开发技术

2.3.1 web 应用框架 Flask

Flask 是由 Armin Ronacher 受 Sinatra Ruby 启发开发的基于 python 的轻量级 web 框架，框架的最大特征是轻便，旨在为用户的各种应用程序提供坚实的基础，框架本身相当于一个内核，只包含基础核心功能，不包含表单验证、数据库抽象层等其他 python 框架具有的功能，但是 Flask 支持各种各样的外部扩展来帮用户实现这些功能，例如邮件使用的 Flask-mail, 缓存使用的 Flask-cache, 用户认证的 Flask-Login, 以及本系统中使用的对象关系映射工具 SQLAlchemy。

Flask 基于 Jinja 模板引擎和 Werkzeug WSGI 工具包，两者均由 Flask 设计者所在团队研发。Jinja 是一种流行的 python 模板引擎，负责将模板与特定数据源

组合实现动态网页呈现，其模板文件存储在 Flask 中的 templates 文件中，WSGI 全称为 Web Server Gateway Interface(Web 服务器网关接口)，是 python 应用中 Web 服务器与框架或 Web 应用程序质检定义的通用接口规范，以保障不同 web 服务器和不同 python 应用程序之间的相互通信。Werkzeug 是基于 WSGI 的工具包，实现了请求、响应和实用函数，为 Flask 框架的构建提供基础。

技术优势：和 Django 等其他成熟 python 框架相比，Flask 虽然年轻但具有独有的优势和良好的发展前景。Flask 的轻量级和模块化设计，以及丰富的扩展工具支持使其具有简单可扩展的特点。其内置开发服务器和快速调试器，并支持应用程序集别和库级别的单元测试，方便开发者在本地环境中快速调试。

2.3.2 分布式任务队列 Celery

Flask 框架为阻塞式框架，即 Flask 一次只能处理一个请求，多个请求到达时，flask 会按序依次执行。当请求中存在耗时较长的 io 操作或复杂运算时，后序的请求都要排队等待，这对需同时响应多个请求的应用服务器来说是不可容忍的，由于本系统中会涉及批量度量等耗时操作，所以使用 Celery 分布式任务队列处理异步任务调度。Celery 中主要包含三个组件，分别为生产者 (Celery Client)，消费者 (Celery Workers) 和消息传递者 (Message Broker)，工作的基本流程如图 2.3 所示：生产者指的是发送任务请求的用户或者应用，在 Flask 框架中生产者

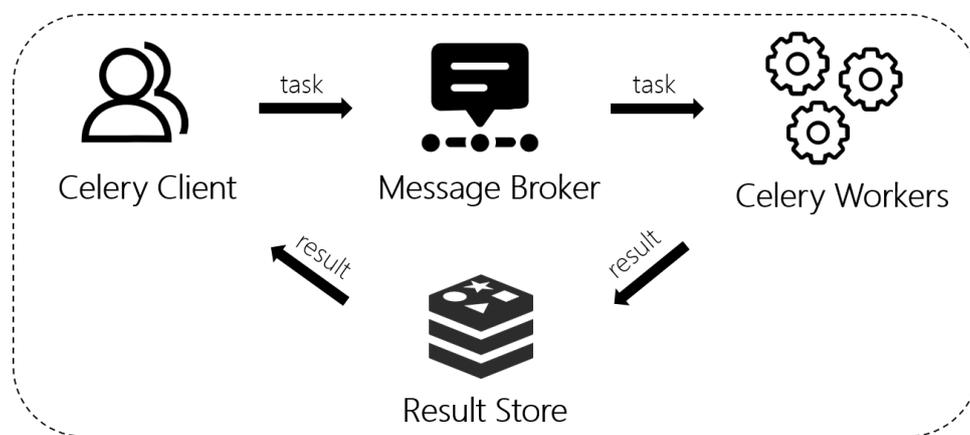


图 2.3: Celery 基本工作流程图

在 Flask 应用中运行，将应用中产生的异步任务交给消息传递者，消息传递者是生产者和消费者之间的交互使者，用于存储应用产生的任务即先后顺序，Celery 本身不提供消息服务，但是支持 RabbitMQ 及 Redis 等多种方式的中间件，本系统中使用 Redis 作为消息中间件。

Celery 会将消息队列中的任务分发给消费者，消费者即任务的处理器，负责

任务的后台运行，Celery 可以设置消费者的个数，可以在本地调试时选择单一消费者进行测试，在生产环境中添加更多的消费者加快后端处理任务速度。为了形成闭环，任务处理完成后，消费者会将运行结果存储至 task result store，应用可以根据需求从仓库中调取任务运行结果，Celery 支持多种类型的 Backend，本系统中使用 Redis 作为支撑。

技术优势：Celery 使用简单，熟悉了 Celery 的工作流程后，配置好依赖环境即可使用；高效：一个单进程的 Celery 每分钟处理任务可达百万；高可用性：当任务执行过程中发生中断或者执行失败，Celery 会自动尝试任务重启；高扩展性：Celery 中几乎每个组件都可以定制或扩展。

2.3.3 对象关系映射工具 SQLAlchemy

SQLAlchemy 是基于 python 实现的 ORM 框架，为用户提供关系对象到数据库的映射，极大的简化了开发人员在原生 SQL 的操作，让开发者将主要精力投入在代码逻辑中，并且可以避免 web 应用中常见的 SQL 注入攻击，SQLAlchemy 包括 SQLAlchemy Core 和 SQLAlchemy ORM 两种使用模式，用户可以根据需求使用其中一个或一起使用。Core 模式与传统 SQL 类似，关注数据表、键值和索引，适用于数据仓库、报表及分析等大量数据甚至多个数据库的数据。ORM 模式关注领域驱动设计，将元数据和业务对象的底层结构和模式进行封装，简化开发者的数据库交互过程。ORM 建立在 Core 之上，Core 和用户选择的 DB-API 对接，实现和底层数据库的对接。

技术优势：代码理解性强，对使用者隐藏了底层的 SQL 语句，提高了开发者的开发效率，同时，SQLAlchemy 能够兼容多种类型底层数据库，包括 SQLite、MySQL、Oracle、MS-SQL、SQLSever 等常用数据库，并且为其他关系型数据库提供添加支持的方式。

2.3.4 Restful 接口测试工具 Flasgger

Flasgger 是 flask 框架下支持的 Swagger，方便用户调试 webAPI 的接口。Swagger 是一种接口文档自动生成和接口测试工具，通过使用 OPENAPI 规范的接口及接口相关信息格式定义接口，再通过衍生的工具生成各式各样的接口文档及各种语言的客户端和服务端代码，实现在线接口调试页面，方便用户快速调试，在应用进行版本迭代时，只需要更新 Swagger 的描述文件即可自动重新生成接口文档和测试代码，保障了接口文档、客户端代码和服务端代码的一致性。

在使用 `flasgger` 时，前后端需要先商定接口文件，包括请求类型、内涵信息及信息格式，以及错误响应的处理等，`swagger` 提供了可视化编辑器 `swagger-editor`，用户可以在交互式环境中编辑接口文档，文件保存为 `yml` 格式文件，后端在进行接口测试时使用 `@swagger_from` 声明接口文件地址，自动生成服务器端测试代码，模拟前端发起请求。

2.4 本章小结

本章简要介绍了本文中涉及到的技术及研究方向的相关内容，主要分为裁判文书质量评估相关研究综述，自然语言处理技术综述以及系统使用开发技术三部分。第一部分包括数据质量研究和信息质量研究两方面，为本文评估指标的设计提供理论依据，第二部分包括文本风格研究、文本相似度计算研究以及 BERT 发展综述，对本文中用到三种自然语言处理技术进行了介绍；最后一部分涵盖了系统实现过程中需要用到的开发技术，包括开发框架 `Flask`、分布式任务队列 `Celery`、对象关系映射工具 `SQLAlchemy` 以及接口测试工具 `Flasgger`。

第三章 民事裁判文书质量评估指标设计

3.1 客观质量评估指标设计

为了体现司法公平性，裁判文书作为记录案件的重要载体，需要具有和普通文本不同的程式化，实录性的语言特征，即对于不同的案件内容和判决结果都具有相同的框架结构，这也是我们从客观角度度量裁判文书的理论依据。客观质量评估指标借鉴了客观信息质量度量传统的数据质量度量方法，分别从细致性、延迟性、真实性、完整性、一致性、易读性以及准确性七个度量维度出发，以最高院发布的《人民法院民事裁判文书制作规范》《民事诉讼文书样式》为格式范本，根据 2019 年 4 月 18 日法官论坛中提出的《民事裁判文书制作存在的问题》制定了详细的度量方法，具体内容如表格中所示：

表 3.1: 客观信息质量度量表

客观信息度量维度及说明	维度说明	民事裁判文书度量方法
细致性	本体能够分解成颗粒的粗细程度	参诉人信息细致性 事实部分细致性 理由部分细致性
延迟性	反应时间和状态发生时间之间的延迟长短	案件信息延迟性 裁判文书延迟性
真实性	载体状态对本体状态的还原程度	案由信息类别规范性 裁判依据引用规范性

表 3.2: 客观数据质量度量表

数据度量维度及说明	维度说明	民事裁判文书度量方法
完整性	实际数据与所期望数据在数量上的满足程度	判决内容说明完整性 诉讼费承担说明完整性
一致性	在一定条件下，数据之间满足的相同条件或状态	数字使用一致性 标点符号使用一致性
易读性	实际数据表达信息的了解或理解的难易程度	事实描述部分的简明性 争议焦点条理性
准确性	实际数据与所期望数据在格式上的一致性	构成事项顺序性 审理经过顺序性 参诉人信息准确性

3.1.1 客观信息质量度量指标设计

3.1.1.1 细致性

信息的细致性度量来源于信息熵与信息粒度粗细的关系，研究表明，信息粒度越大所包含的信息价值越低，反之，信息的粒度越小所包含的信息越多。由于我们的数据是处理后的 xml 文件，廉昊在论文中提出通过度量 xml 中树的平均深度衡量数据的细致性，但实际上 xml 文书中含有许多固定格式标签，例如文首，诉讼记录等，也存在文书包含事项不同，例如有的文书包含附件而其他文书不具备。统一度量忽略了这些因素对细致性度量结果造成的影响，所以论文对细致性度量的信息做了细化，分为以下三点：

参诉人信息细致性

参诉人主要包括当事人和诉讼代理人，设案件中涉及的当事人数量为 m ，每个当事人解析后得到的 xml 标签的所有子标签数量为 L_i ，代理人数量为 n ，代理人信息解析后得到的 xml 标签的子标签个数为 L_j ，该裁判文书细致性的定义为：

$$Meticulous_{CSRXX} = \frac{L_i \times m + L_j \times n}{m + n}$$

事实部分细致性

事实部分是案件详细情况的说明，其描述内容是否详细对案件的判断有重要的影响。事实部分在 xml 中对应标签为“案件基本情况”，所以事实部分的细致性为“案件基本情况”标签对应的所有子标签个数 L_f ，计算公式为：

$$Meticulous_{AJJBQK} = L_f$$

理由部分细致性

理由部分为法官的判决分析过程，说理的详细性影响了裁判文书的可信程度。理由部分在 xml 中对应的标签为“裁判分析过程”，所以理由部分的细致性为“裁判分析过程”标签对应的所有子标签 L_r ，计算公式为：

$$Meticulous_{CPFYGC} = L_r$$

3.1.1.2 延迟性

延迟性在客观信息论中的定义为反应时间对状态发生时间的延迟，这和数据质量中提到的及时性相似，数据记录的时间和实际产生的时间间隔越短，其

中包含的信息损失的程度就越少，裁判文书中存在多个时间点，我们根据实际情况将延迟分为两个阶段：

案件信息延迟性

案件信息的延迟指的是从案件发生到立案日期之间的延迟，案件发生定义为参见描述中最早提及的日期 T_{start} ，立案日期是指人民法院对上诉人的申请材料进行审查后，接受其上诉申请并开展进一步调查和审判活动的日期，设定为 T_{accept} ，则案件从发生到进入法律流程之间的延迟定义为：

$$Delay_{AJXX} = T_{accept} - T_{start}$$

裁判文书延迟性

裁判文书的延迟性是指立案到审判之间的延迟，在法院接受上诉人的诉讼申请后，需要对案件的发展经过和证据等开展核实和审查，经过开庭等审判程序后得到最终的审判结果，也是裁判文书完成的日期，设为 T_{end} ，则裁判文书延迟性的定义为：

$$Delay_{AJXX} = T_{end} - T_{accept}$$

3.1.1.3 真实性

信息作为本体状态到载体状态之间的映射，是否真实由很多因素决定，这与数据质量领域的可信度相似，而裁判文书中的真实性或可信度主要体现在其专业性，也就是裁判文书中对于专有名词和法律的引用必须准确，主要分为以下两个方面：

案由信息类别规范性

案由是确定案件类型的关键标志，是裁判文书不可缺少的标签信息，并且案由名称需根据最高院给出的《民事案件案由规定》（以下简称《案由》）进行规范，不可胡乱编造。本论文中使用的案由是在 2006 年第一版《案由》的基础上，根据最高人民法院 2018 年发布的两版增加案由的通知修改后的完整案由，共 10 大部分，43 个种类，每个种类最多内涵 4 个层级，合计 424 个案由。系统中案由信息的规范性定义为案由信息是否包含在案由列表中，如不包含，则视为编造案由，影响了裁判文书的可信度。

裁判依据引用规范性

裁判依据是指人民法院判决依据的程序法和实体法条文，是裁判文书说理的根本，所以引用法条的准确性和权威性直接影响了裁判文书的可信度，依据

《民事诉讼文书样式》(以下简称《样式》)要求,“裁判文书不得引用宪法和各级人民法院关于审判工作的指导性文件、会议纪要、各审判业务庭的答复意见以及人民法院与有关部门联合下发的文件作为裁判依据”,且“指导性案例不作为裁判依据引用,只能作为参照引用”,所以裁判依据的引用规范性检测定义为引用内容中不包含指导性案例、会议纪要、行政规章、地方规章、地方政策性规定等非规范性文件。

3.1.2 客观数据质量度量指标设计

3.1.2.1 完整性

数据质量中对于完整性的定义为实际数据在数量上对期望数据的满足程度。裁判文书中分项较多,并且每个分项又可进一步细分,由于案件实际情况不同,每篇裁判文书都可解析成不同的树形结构,但根据《样式》规定,有一些固定的成分只可简化不可缺少,本文对其中重要的组成部分的完整性进行了度量。

判决内容说明完整性

裁判文书作为法院执行判决结果的书面依据,判决内容说明是否清晰完整直接影响了裁判文书的可执行性。民事案件的裁判结果多涉及给付或承担责任两种形式,根据样《样式》说明,对于给付事项,在判决主文中要写明种类、名称、数量、型号和时间,涉及利息的也要明确标注利息计算的起始时间和截至时间。并且在尾部要有“如果未按本判决指定的期间履行金钱给付义务,应当按照《中华人民共和国民事诉讼法》第二百五十三条的规定,加倍支付延迟履行期间的债务利息”的告知说明。

针对以上要求,本文抽取 xml 文书中判决结果标签字段,首先判断是否涉及判决金额,对于涉及判决金额的案件,按照金额、金额类型、判决执行期限及义务人的顺序检查金额说明的完整性和顺序性,检测流程同参诉人信息准确性检测类似,同时按照规则匹配检测判决主文中是否说明告知事项,告知事项是否准确。

诉讼费承担说明完整性

诉讼费即诉讼费用,指的是当事人进行各类诉讼时依照法律规定,需要向人民法院缴纳的费用。主要包括案件受理费和其他费用两种类型。《样式》规定民事裁判文书应在尾部说明诉讼费用的承担情况,格式为“案件受理费.....元,由...负担”,对于涉及多名被告的案件,还应详细说明由每一方承担的诉讼费金额。

评估的具体流程如图3.1所示,首先抽取 xml 中诉讼费承担记录,如果不包含该标签,则记录错误信息“缺少诉讼费承担记录”。承担记录的解析结果中包含诉讼费总金额及诉讼费承担分组,承担分组中包含了承担人的信息,当涉及多

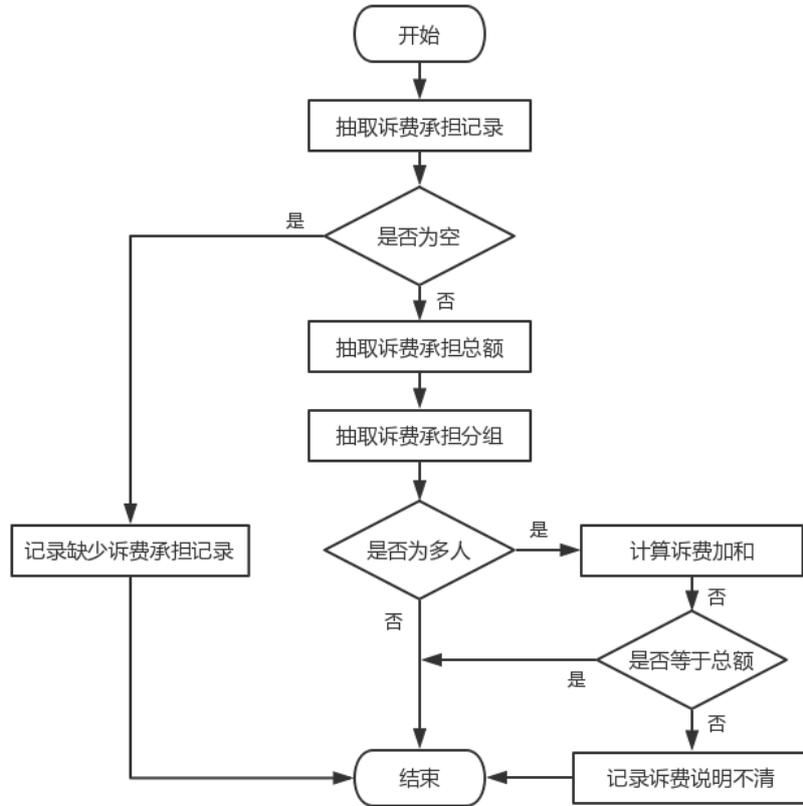


图 3.1: 诉讼费承担说明完整性流程图

3.1.2.2 一致性

个承担人时，如果没有明确个人的承担范围，解析软件会将总金额解析为个人承担金额，所以检测系统通过核查总金额和分组中个人承担金额的加和关系，确定告知事项是否详细说明每个承担人的承担范围，如不等则记录警告信息“未分别说明承担人各自承担金额”。

数据质量度量中的一致性指的是数据是否遵从一定的规范。该维度是从大数据的角度出发，衡量数据之间格式是否统一。裁判文书描述的案件不同，但书写过程中都要遵循统一的格式，除了结构上的要求，还包含内容书写中需要遵循的规范，本文归纳了《样式》中对于数字和标点符号使用的要求，以此作为裁判文书一致性的度量标准。

数字使用一致性

裁判文书中主要使用两种形式的数字，一种是汉字数字，如一、二、三……，一种是阿拉伯数字，如 1, 2, 3……，根据《样式》规定，汉字数字主要出现在以下几种情况：

1. 裁判主文中对判决结果进行分项阐述时；
2. 裁判分析过程中进行分项评述时；
3. 尾部的落款时间

除了以上场景，文书正文和标题中多使用阿拉伯数字，裁判文书中的数字使用格式一致性应严格按上述标准评估。

标点符号使用一致性

早在 1993 年 1 月 1 日试行的《人民法院民事裁判文书制作规范》中就提到裁判文书中正确使用标点符号的重要性和必要性，但实际使用中各级人民法院依然各自为政，造成裁判文书表意不清。本文根据目前施行的《民事诉讼文书样式》中对标点符号的明确规定，结合各级人民法院在文书评查中发现的错误，对关键及易错标点符号使用的规范做了总结和归纳。主要要求的标点符号有逗号、冒号、分号以及顿号，涉及到的详细场景和要求如下表所示：

表 3.3: 民事裁判文书书写过程中的标点符号规范表

标点符号	要求说明	涉及事项名称	xml 文件对应标签
逗号	“被告辩称“后 “本院认为“后	事实 理由	案件基本情况 裁判分析过程
冒号	“原告“、“被告“、“委托诉讼代理人“后 “xxx 向本院提出诉讼请求“、“本院认定如下“后 ” 判决如下“、“裁定如下“后	事实 事实 裁判主文	当事人 案件基本情况 判决结果
分号	裁判项“一、“后,“二、“之前	裁判主文	判决结果
顿号	裁判项序号之后	裁判主文	判决结果

本文首先按照 xml 对应标签抽取出涉及到的文本内容，然后使用规则匹配定位到“原告”等关键字，再对关键字前后的标点符号进行筛选，对于没有按照规范使用的标点符号，输出原文位置及对应的符号修改意见。

3.1.2.3 易读性

易读性是从数据表达特性的角度对数据进行度量的一种方法。裁判文书网上公开的主要意义在于增加司法领域透明度，保障公民的知情权和参与权，同

时鼓励公民行使监督权，所以裁判文书是否易读也是裁判文书书写过程中应该注意的问题。对于易读性，我们主要从条理清晰性和简明性两个角度进行度量，具体度量方法如下所示：

事实描述部分简明性

事实描述部分在 xml 结构中对应的标签为案件基本情况，其中包括原告上诉段、被告辩称段、查明事实段等不同角度对案件情况的概述。裁判文书事实描述部分的书写需参考当事人递交的上诉申请书等材料，但很多文书存在原文照抄，不加整理归纳就使用，导致当事人诉辩的事实和理由部分过长。所以本文通过检测裁判文书上诉段和辩称段的长度，定义文书事实描述部分的简明性。

争议焦点条理性

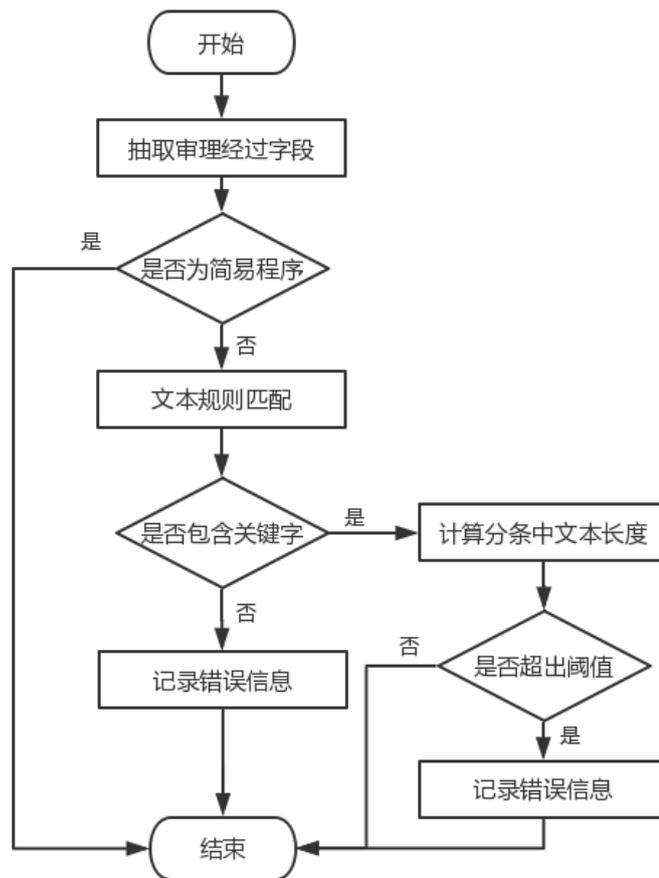


图 3.2: 争议焦点检测流程图

案件上诉的原因主要在于原告与被告存在争议，需要司法机关介入裁决，所以案件的焦点应该是案件核心部分，《样式》中在理由部分的制作规范提到，“除非常简单案件或当事人全部承认案件事实的以外，一般都应归纳争议焦点”，并提到“归纳争议焦点应当明确具体，争议焦点之间应当具有层次性于逻辑性”。

本研究根据规定按照流程图3.2进行争议焦点检测。首先根据审理经过中的一审案件适用程序，将案件分为简易程序案件和普通程序案件两类分别进行处理。对于简易程序的案件不进行焦点检测，默认其为“非常简单案件”；对于普通程序案件，使用规则匹配的方式，对裁判文书的裁判分析过程进行检测，测试其是否包含“本案的争议焦点为“或”本案焦点“等总结字样，并对其后分条的焦点长度进行检测，检测结果定义为裁判文书争议焦点条理性的评估结果。

3.1.2.4 准确性

在数据质量度量规范中，准确性指的是实际数据与所期望数据之间的一致程度。为了保证裁判文书格式的统一，《样式》对其中的一些结构组成和顺序做了明确的规定，司法相关人员在书写的过程中需严格满足模板要求，我们选取了构成事项，审理经过及参诉人信息三个重要信息项进行了准确性的度量。

构成事项准确性

表 3.4: 民事裁判文书中构成事项及说明表

事项结构	事项名称	事项内容说明	xml 结构对应标签
标题	标题	由承办法院名称，文书名称和案号组成	文首 (WS)
正文	首部	包括诉讼参与人信息，案件由来及审理经过	当事人 (DSR) 诉讼记录 (SSJL)
	事实	包括当事人的诉讼请求，原告和被告的诉辩内容，以及人民法院认定的证据及事实	案件基本情况 (AJJBQK)
	理由	人民法院根据原告和被告提供的案件事实，结合相关的法律依据对双方的争议焦点进行分析，并阐明理由	裁判分析过程 (CPFUGC)
	裁判依据	法院作出最终判决所依据的实体法和程序法	法律法条引用 (FLFTYY)
	裁判主文	人民法院对案件最终做出的明确，具体的处理	判决结果 (PJJG)
	尾部	对诉讼费用承担和其他告知事项	诉讼费承担 (SSFCD) 等
落款	落款	裁判文书书写人的署名和日期	文尾 (WW)

据《样式》说明，民事裁判文书主要分为标题，正文和落款三部分，正文中又包括首部、事实、理由、裁判依据、裁判主文和尾部，其中每个事项说明如下，事项中内容可以根据实际情况适当减少，但事项本身必不可少，并且必须按照规定的顺序依次说明，以此衡量裁判文书构成事项的准确性。

审理经过准确性

审理经过是对整个案件由受理到最终结案的完整过程的概述，相当于裁判文书的摘要，所以案件关键的流程节点信息缺一不可，审理经过应包括但不限于案件受理时间、开庭时间、适用程序、到庭情况、审理结果等，以此衡量审理经过的准确性。实际的处理流程与参诉人信息完整性流程相似。

参诉人信息准确性

参诉人是案件涉及到的所有人员，包括起诉方、应诉方、代理人三种角色，不同的角色所需要的信息也不尽相同，根据《样式》规定，具体要求如下：

表 3.5: 民事裁判文书中对参诉人信息记录的要求说明表

当事人身份	记录信息说明
自然人	基本信息应按照姓名、性别、出生年月日、民族、籍贯、住址、公民身份证号码书写，并且应做到尽量详细
诉讼代理人	当事人有法定代理人或指定代理人，应另起一行以“委托诉讼代理人”开头写明姓名、性别、职业、工作单位、住所等信息，多个代理人的不可合并，事务所的法律工作人员应写为“法律工作者”，不可简称为“律师”
法人组织	法人组织的名称以注册时登记的文件记载为准，住所为法人或其他组织的主要办事机构，办事地不明确的以注册地为准。
个体工商户	名称为经营者的姓名、性别、出生年月日，住址以身份证或营业执照上登记的为准
个人合伙	未起字号的，当事人为全体合伙人；起字号的需注明与合伙人之间关系，住址同个体工商户要求
其他组织	名称均以注册时等级文件记载为主，住址同法人要求

具体度量流程如图3.3所示，系统首先从 xml 文件中抽取当事人标签，并对其子标签中 tag 为“起诉方”，“应诉方”及“代理人”的当事人个体依次进行检查。对于每个当事人，首先按照当事人的类型设定标准的信息项序列，然后按照标签筛选得到当事人的信息项，例如当事人的姓名、性别、出生年月日等，接着按照标签内的信息到原文本中匹配位置，最后按照匹配得到的位置序列和标准序列进行比对，输出缺少或者顺序错误等检测结果。

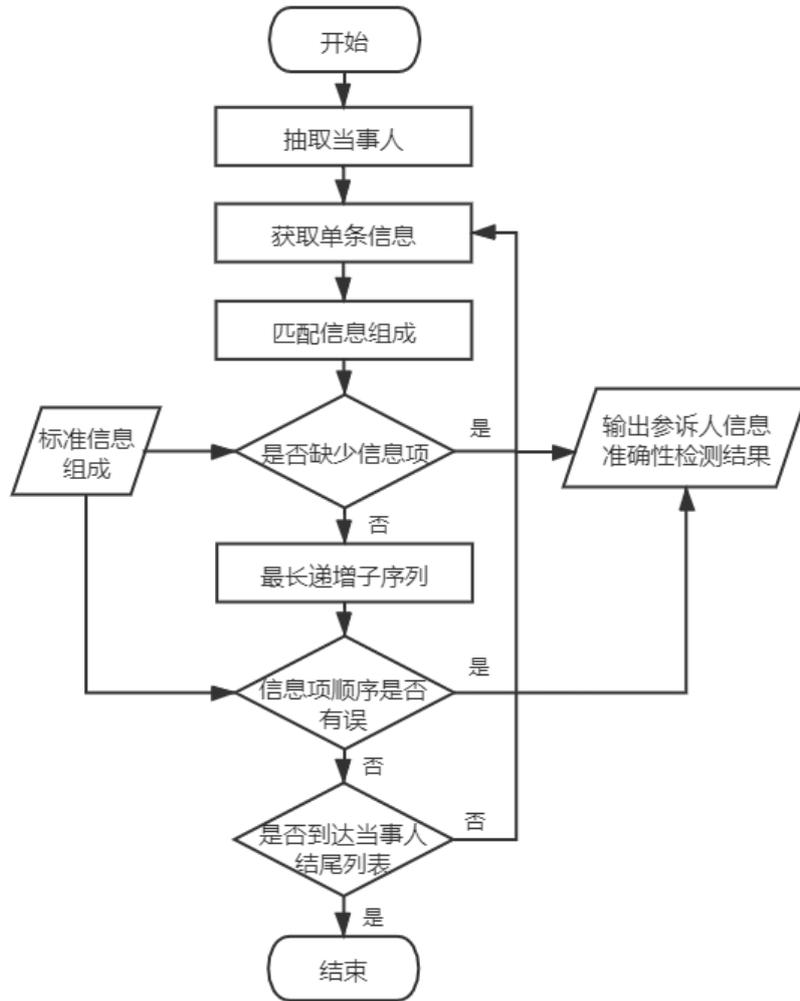


图 3.3: 参诉人信息准确性流程图

3.2 主观质量评估指标设计

3.2.1 引用法条合理性度量

由于各级法院的法官们年龄、知识体系和审判经验上存在差异，对于法律的理解和运用方法各不相同，造成同一个案件在不同地点，不同法官的判决结果存在差异的情况。为了弥补人力的不足，本系统借助计算机的大数据计算能力，使用人共智能的方法对已有的裁判文书的法条引用合理性和全面性进行评估。评估方法采用基于相似案件协同过滤的评估。模拟法官查询和当前案件相似的样例案件，然后参考样例案件依据的法条给出本案件的依据。我们使用当前案例中引用的法条和相似案件引用法条的重合度，作为文书引用发条合理性的度量结果。

3.2.2 语言风格度量

裁判文书是对案件重要流程节点的纸质记录，也是法院行使司法决策的依据，所以其在语言风格和结构上区别于一般报告、新闻等社会性文本，具有文辞规范性、表述客观性、逻辑条理清晰性等语言特征。通过对现有的裁判文书调查显示，许多文书存在照抄指控事实，机械套用样式模板，语言表述不规范等问题，而这种问题的发现多依赖专家对裁判文书的人工筛查，这种方法不适用于大数据时代海量的裁判文书质量评估。本文使用自然语言处理领域的技术方法，实现对文本的自动文本风格评估，具体方法如下：

3.2.2.1 语言风格鲜明性

无论是文书中的原告诉称段还是对被告辩称，都需要注意使用词语的专业性和评述的客观性。编写者在转述时应当尽量使用“法言法语”，这不但是对文书编写者语言运用能力的考察，也是司法工作者转换思维的体现，即从法律问题去观察、思考和评判问题，通过法律术语的使用让读者迅速抓住案件关键。评述性语言使裁判文书在叙事时的关键特征，要求语气，口吻应当是客观中立，不包含任何感情色彩的，只有围绕法律规范的要素对案情进行客观的叙述，才可作为法官评判分析的依据。

为了区别裁判文书中陈述语言和新闻文本语言的风格，本文以裁判文书和司法类新闻作为训练数据，使用基于卷积神经网络的分类算法训练分类器，捕捉两种文本之间存在的文本风格差异，最终以模型分类的评分作为待测文书语言风格评估的得分。

3.2.2.2 语言情感客观性

判决文书作为司法公正的书面载体，其语言须客观中正，不能偏颇任何一方当事人，所以裁判文书正文中的每一句话，都应该持中立的立场，尤其是文书中涉及到事实描述的段落，更应当理性客观的阐述案件发生的经过和结果，原告被告的诉求如何。所以本文对 xml 文书中的案件基本情况标签下的文本进行语言情感客观性评估，评估方法主要有两种，第一种是使用情感词典，对句子进行分词处理后，借助极性词典对句子的客观性进行打分；第二种是使用带标记的数据集训练深度学习的分类模型，对于待测数据使用训练好的模型进行分类预测，将预测结果作为文本情感客观性的评分。最后将两种算法的结果相融合作为句子最终的情感客观性得分。

3.2.2.3 文本重构性检测

查明事实段时对法院审理查明过程的概述，是与最终判决结果有直接关系的全部案情事实。本段应该在原告被告诉辩内容的基础上，结合法院、公安局

等各权威机构查证的结果，对案件的涉案人信息，发展经过，情节状况，查明证据进行完整详细的阐述，而裁判文书的质量调查和抽样显示表明，为了节省时间和人力，部分裁判文书存在照抄原告诉称及被告辩称段，并未对案件情况进行内容重构，因此本文对文书中的查明事实段和原被告告诉称段进行抄袭检测，以检测的相似度结果作为语言重构检测质量的评估指标。

3.3 本章小结

本章主要介绍了民事裁判文书质量度量指标的设计，分为客观质量评估指标设计和主观质量评估指标设计两部分。客观质量度量主要考虑文书浅层信息规范性，借助信息质量度量 and 数据质量度量的理论方法，从细致性、延迟性、真实性、完整性、一致性、易读性、准确性七个角度出发，设计了 15 项详细的度量指标和对应的计算方法。主观信息质量试图衡量文本深层的语义信息质量，借助自然语言处理技术，实现裁判文书中关键文本的语言风格鲜明性、语言情感客观性，文本重构性以及裁判文书引用规范性的度量。

第四章 民事裁判文书质量评估系统需求分析及设计

4.1 系统边界

本系统研究内容隶属多元智能化诉讼服务及审判执行关键技术课题，为构建司法基本服务库中的基本服务之一，和其他系统关系及业务关联如下图所示，其中实线框内为本系统的质检服务范围：

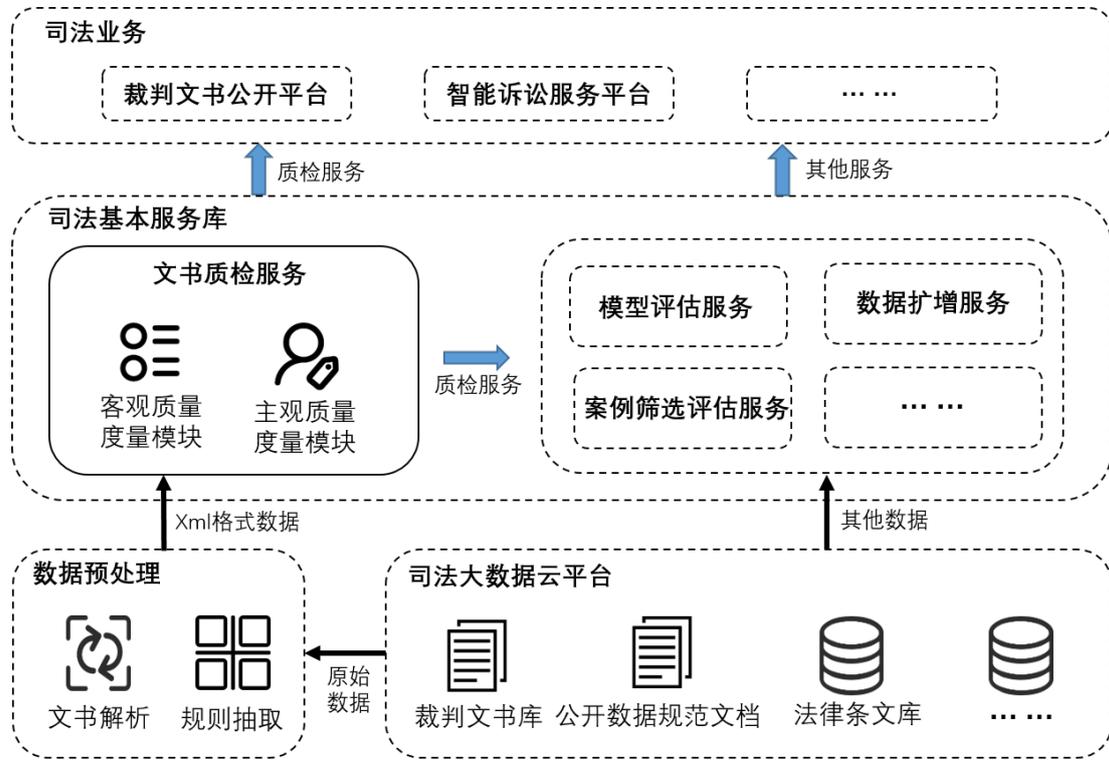


图 4.1: 系统边界图

司法大数据云平台中积累了海量司法相关数据，其中包括审判执行数据、司法研究数据、信息化管理数据和司法政务数据、信息化管理数据及外部数据等，也是各个上层业务系统实现的数据基础，本系统主要涉及到的数据包括裁判文书数据集、公开数据规范文档以及法律条文库。

在进行数据质量检测服务之前需要经过数据预处理模块，其中文书解析由第三方解析工具完成，将原始裁判文书按规则解析为半结构化 xml 文书格式，为质检服务做准备。规则抽取是指通过规则匹配等方法，将公开数据规范文档中

的文本内容转换成结构化数据，作为度量方法判决依据的一部分。预处理后的文件输入文书质检服务。

司法基本服务库是底层数据与上层业务系统之间的枢纽，为上层的裁判文书公开平台，智能诉讼服务平台等其他司法业务提供质量检测，案例筛选等基本服务。文书质检服务为司法基本服务库的一部分，内含客观质量度量和主观质量度量两大模块，对外提供定制化质检服务，实现对已公开文书的质量评估和未公开文书的质量筛选。基本服务库中还包含模型评估服务，数据扩增服务及案例筛选服务，对数据扩增服务，本系统提供扩增前后数据的质检服务，帮助过滤扩增产生的低质量数据，对模型评估服务，本系统提供评估指标及对应的筛选结果，观察不同质量评估指标下数据质量对模型质量的影响。

除此之外，由于本系统需要作为公司质量测试平台的子平台，除了提供质量检测服务接口以外，还应为前端提供用户数据管理接口和可视化报告数据接口。综上所述，裁判文书质量评估平台的总体功能如下

1. 文书上传及质检结果下载：系统为用户提供单篇或批量上传文书接口，存储到文件系统后将文书信息持久化到数据库中方便后期使用。检测报告结果提供 json 格式，方便用户下载及前端页面展示。

2. 用户数据管理：为用户提供上传文书管理和批量检测任务管理，用户可在前端对文书列表和任务列表实现增删改查基本操作，后端提供对应接口相应前端请求。

3. 质量检测服务：系统提供可自定义指标组合的文书质量检测服务，包括基于规则的客观质量度量和基于自然语言处理技术的主观语义质量度量。

4. 统计分析服务：对于批量数据质检服务，在单篇质检服务的基础上提供统计分析处理服务，方便用户了解文书集整体质量情况。

4.2 系统需求分析

本小节在前文系统边界划分的基础上对系统需求进行详细介绍，主要分为用户数据交互及管理 and 文书质量检测与统计分析两部分，具体内容如下：

4.2.1 用户数据交互及管理

如图4.2所示，用户数据交互及管理包括数据交互、配置管理及数据管理三部分。

数据交互主要包含文书上传及质检结果下载，是评估对外服务的入口和出口，为文书的质检服务提供数据基础，其中文书上传又包括单篇文书和批量文书

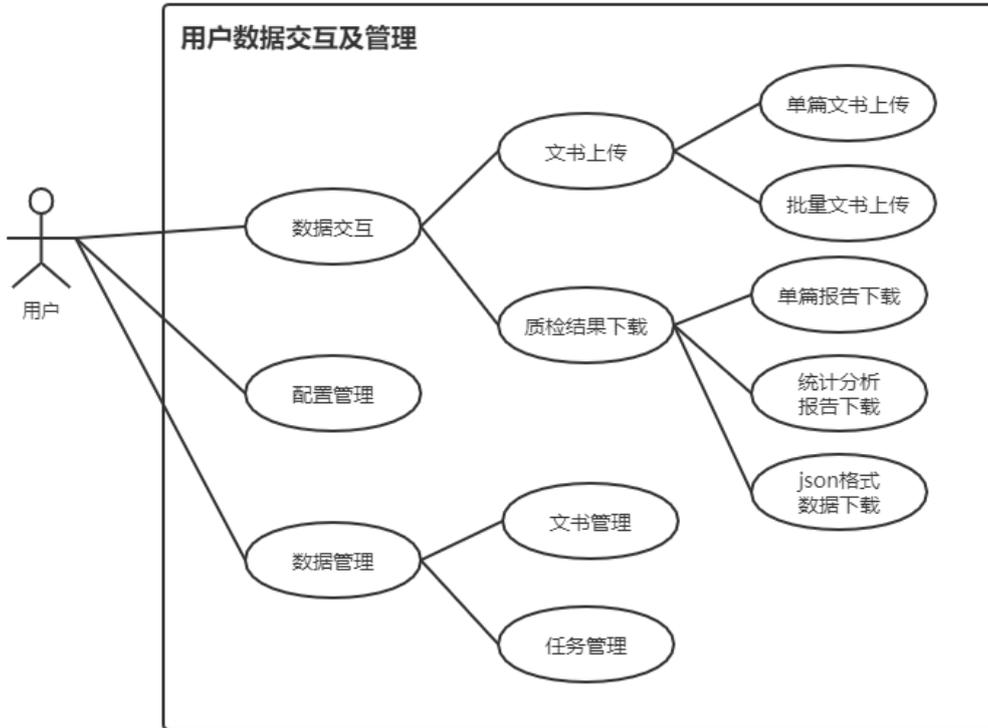


图 4.2: 用户数据交互及管理模块用例图

上传，满足普通用户的单篇文书质检请求和企业用户的批量检测需求，表4.1提供了用户使用文书上传功能的用例描述。

表 4.1: 用户数据上传用例描述表

UC1	用户数据上传
参与者	用户，目的为上传单篇或多篇文书
触发条件	用户点击“文件上传”按钮，前端像后端发送文件上传请求
前置条件	用户已登录，并判断用户上传文书总数是否超过限定值
后置条件	将用户上传的文书存储到后端数据库，并将文书对应的信息持久化到数据库中
优先级	高
正常流程	<ol style="list-style-type: none"> 1. 用户进入我的文书界面，点击文书上传按钮 2. 用户在本地选择单个文件或者文书压缩包进行上传 3. 后端判断上传文件格式是否符合要求，将符合要求文件进行存储。 4. 将存储成功的文件信息存储到数据库，并将结果返回给前端 5. 前端显示用户上传成功文件列表。
特殊需求	用户上传前提示上传文书信息格式，并将上传成功列表返回给用户

表 4.2: 质量检测结果下载用例描述表

UC2	质量检测结果下载
参与者	系统用户, 目的是下载质量检测结果文件留存
触发条件	用户点击下载报告或下载统计报告按钮
前置条件	单篇报告下载: 文书质量检测完毕, 统计报告下载: 任务内全部文书质量检测完毕且统计分析过程完成
后置条件	无
优先级	高
正常流程	1. 用户进入报告查看界面 2. 用户选择下载报告格式并点击下载按钮 3. pdf 格式报告直接在前端生成, json 数据格式从后端获取 4. 用户查看下载结果。
特殊需求	用户下载 pdf 格式报告时应将报告中折叠部分全部展开

质量检测完成后, 除了前端使用可视化图表的形式展现给用户之外, 系统还提供质检结果下载服务, 方便用户留存作为质检依据, 或者进行后续的分析处理。下载支持用户两种方式, 用户可选择可视化报告结果直接下载, 即 pdf 形式, 也可选择后端数据格式查看各项指标数值结果, 即 json 格式, 以便后续自定义的分析处理, 对于批量检测的数据集, 还支持用户下载两种格式的统计分析结果, 具体用例描述如表4.2所示。

表 4.3: 用户配置管理用例描述表

UC3	用户配置管理
参与者	系统用户, 目的是修改默认指标配置
触发条件	用户点击修改配置按钮, 前端记录用户提交表单内容后向后端发起修改请求
前置条件	用户进入我的配置页面, 修改配置后点击保存配置按钮
后置条件	用户在配置后系统会默认使用用户自定义的配置对文书进行质量检测
优先级	高
正常流程	1. 用户进入我的配置界面查看当前配置 2. 用户点击修改配置按钮, 选择指标后点击保存修改按钮 3. 后端收到请求后修改数据库中保存的用户配置信息, 将修改结果返回给前端 4. 前端提示用户是否修改成功。

系统的管理功能主要分为数据管理和配置管理两方面, 配置管理为用户提供自定义指标配置功能, 用例描述如表4.3所示, 用户可以根据自己的需求随时更改质量度量的指标组合, 系统会默认按照用户的配置对文书质量进行度量, 质

检过程中使用的检测项也会显示在质检报告中。

数据管理包括文书管理和任务管理，文书管理支持用户查看已上传文书列表，或按照关键字筛选文书列表，用例描述如表4.4所示。除此之外，为了满足用户批量检测的需求，支持用户选择多篇文书创建任务，并对任务进行基本的操作处理，具体用例描述如表4.5所示。

表 4.4: 用户文书管理用例描述表

UC4	用户文书管理
参与者	用户，目的是管理已上传文书
触发条件	用户点击我的列表界面，对界面进行删除和检索等操作
前置条件	用户已登录，并且已上传文书
后置条件	用户可以选择单篇文书进行质量检测，或者选择多篇文书创建批量检测任务
优先级	高
正常流程	<ol style="list-style-type: none"> 1. 用户进入我的文书界面，系统像后端发起 get 请求，获取当前用户相关文书，向用户展示文书列表 2. 用户选择关键字或时间区间，点击搜索按钮，系统将过滤条件发送给后端，后端返回过滤结果呈现给用户 3. 用户选择文件点击删除按钮，后端删除文件及相关信息后将结果返回给用户，前端刷新文书列表。
特殊需求	支持文件名的模糊检索

表 4.5: 用户任务管理用例描述表

UC5	用户任务管理
参与者	用户，目的是新建或查看批量质检任务
触发条件	用户选择多篇文书创建批量检测任务，以及用户查看文书列表。
前置条件	用户已登录且已上传多篇文书
后置条件	用户建立批量检测任务后后端会自动对任务中所有文书进行质量检测，检测完毕后生成统计报告存到后台数据库，并修改任务状态。
优先级	高
正常流程	<ol style="list-style-type: none"> 1. 用户在我的文书列表中选择多篇文书后点击创建任务 2. 用户可以为任务单独配置检测指标组合，也可选择一键检测使用默认配置完成检测。 3. 后端自动开启检测任务对任务列表中的全部文书进行质量检测，并自动生成统计分析报告，同时更新文书状态和任务状态 4. 用户对任务列表进行刷新后即可看到所有任务的当前状态。
特殊需求	任务状态栏显示颜色不同，便于用户区分

4.2.2 文书质量检测与统计分析

文书质检与统计分析为用户提供单篇文书质量检测与多篇文书统计分析两个功能，用例图如图4.3所示：

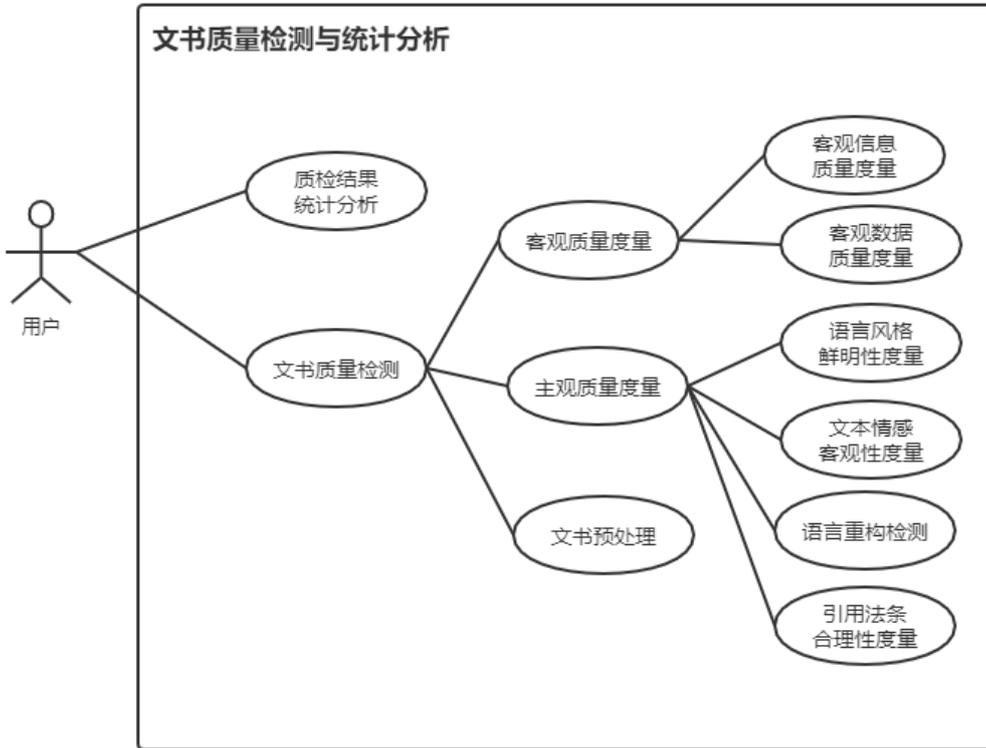


图 4.3: 文书质量检测与统计分析模块用例图

表 4.6: 文书预处理用例描述表

UC6	文书预处理
参与者	系统用户，目的为发起文书质检任务
触发条件	用户点击单篇文书的质量检测按钮或创建批量检测任务
前置条件	用户已登录且已上传文书，文书中存在未检测文书
后置条件	文书客观质量检测 and 主观质量检测
优先级	高
正常流程	1. 用户选择待测文书点击质量检测按钮，或发起批量检测任务 2. 前端将检测请求发送给后端，后端在数据库中找到文件后对数据进行抽取和处理 3. 处理成功后传给质量检测模块进行检测，并将部分标签数据存储至数据库。

对于输入的文书首先要进行文书预处理，这是文书进行质检前必须要经过的步骤，对用户来说为自动触发且不显示解析结果，解析工作主要分为两部分，一是按照规则抽取客观及主观度量需要的标签项，并对涉及到数据的标签项进行数据预处理；二是在前一步基础上对主观度量中需要使用的文本数据进行分句、分词、向量化等文本处理，用例描述表如表4.6所示。

文书质量检测为系统的核心功能，包含文书客观语法与主观语义质量度量两个模块，每个模块下细分多个指标，用户视角对指标并不做划分，用户可以在所有指标间任意选择指标组合完成质量检测任务，具体用例描述表如表4.7所示。

表 4.7: 文书质量检测用例描述表

UC7	文书质量检测
参与者	系统用户，目的为对文书进行质量检测
触发条件	用户发起单篇文书质量检测请求，或创建批量检测任务
前置条件	文书预处理过程完成
后置条件	用户在下次查看时可直接调取报告的 json 格式，通过前端渲染后呈现给用户
优先级	高
正常流程	<ol style="list-style-type: none"> 1. 获取用户配置 2. 根据用户选择的指标项组合调取相应检测方法，完成对文书的客观质量评估和主观质量评估。 3. 将评估结果转换成前端需求格式返回，渲染为可视化报告呈现给用户，同时以 json 数据格式存储到文件系统中 4. 修改数据库中文书检测状态

表 4.8: 质检结果统计分析用例描述表

UC8	质检结果统计分析
参与者	系统用户，目的为对文批量文书数据质量进行分析
触发条件	用户发起文书检测任务
前置条件	任务中所有单篇文书质量检测完成
后置条件	用户在下次查看时可直接调取报告的 json 格式，通过前端渲染后呈现给用户
优先级	高
正常流程	<ol style="list-style-type: none"> 1. 获取用户配置及单篇文书列表 2. 从文书质检结果以及数据库中获得统计分析所需信息项，计算统计分析结果 3. 将统计分析结果转换成前端需求格式返回，渲染为可视化报告呈现给用户，同时以 json 数据格式存储到文件系统中 4. 修改数据库中任务检测状态

为满足企业用户批量检测的需求，方便用户了解数据集整体质量，系统为用户提供了质检结果统计分析服务，对于用户发起的任务检测，系统会在对任务中的单篇文书质检完毕后自动生成统计分析报告，报告中包含文书基本信息的分布统计，以及各指标表现情况分析。具体用例描述表如表4.8所示。

4.2.3 系统功能性需求

系统功能需求主要分为四个模块，分别为数据交互模块，用户数据管理模块，单篇文书质量检测模块以及批量文书统计分析模块，各个模块中功能要求如表4.9所示：

表 4.9: 功能需求分析表

字段	类型	允许为空
数据交互	文书上传	允许用户从本地选择单篇或多篇文书进行上传，并且对上传文书格式进行筛选后存储到后端服务器。
	质检结果下载	支持用户在单篇报告或统计分析报告查看界面下载报告结果至本地，报告格式支持可视化 pdf 版及 json 数据版
用户数据管理	文书管理	支持用户查看已上传文书列表，并可根据关键字对列表进行过滤和检索，方便用户查找相关文书
	任务管理	支持用户选择多篇文书组成批量检测任务，并可在我的任务列表中查看已建立任务列表及每条任务的当前状态
	配置管理	用户可在我的配置中修改全局配置，自定义质量度量指标的组合，用户发起质量检测请求时如果没有定义检测指标，则使用默认配置完成检测
文书质量度量	客观质量度量	支持对文书的客观信息质量度量和数据质量度量，对用户上传的文书给出每个度量项的具体得分和相关解释
	主观质量度量	支持对文书的四个主观语义指标度量，对输入文书给出主观质量得分及每个指标相关解释文本
	文书预处理	系统可根据客观及主观度量指标计算需要，从待测文书中自动化抽取信息项
质检结果统计分析	统计分析	系统提供批量文书检测的统计分析报告

4.2.4 系统非功能性需求

为了保障系统的良好运行，系统在满足用户功能需求的同时，还应满足以下非功能性需求：

1. 可扩展性

系统的指标度量方法之间要尽可能降低耦合度，在使用时采用开关式设计，满足用户随机组合指标的需求。后期需要增加其他指标时，定义度量方法后在 config 文件中配置即可。

2. 可理解性

系统设计的指标应该简单易懂，在用户选择指标时，对于有歧义的指标应该向用户提供解释说明，在质量检测报告中对于涉及到文本内容的指标应该在原文相应位置标记说明，方便读者查看具体解释。

3. 健壮性

由于文书格式和内容不统一，解析软件获得的标签内容不相同，所以系统在度量过程中需要考虑标签内容缺失或错误导致的异常错误，对错误进行处理并将错误原因反馈给用户。

4. 易测试性

由于本论文主要内容为系统后端部分，为了方便和前端对接及模块测试，系统应提供详细接口文件以说明接口使用规范，并且对于度量过程中的具体度量指标，应尽量相互独立方便单独测试。

5. 响应性能

为满足前端界面对大量用户的并发请求操作，后端使用 celery 实现异步任务处理，保证前端界面的快速响应，同时对于度量等耗时处理均采用异步响应形式，减轻主线程计算压力。

4.3 系统概要设计

4.3.1 系统整体框架

系统总体结构如图4.4所示，本系统采用了前后端分离的设计方式，包括前端交互展示层，应用接口层，数据质量度量层，数据处理层，交互调度层及数据存储层，本小节将对系统的整体架构做简要的介绍。

前端交互展示层为用户提供数据交互功能和可视化评估功能。前端基于 VUE 框架，能更好地支持组件化和模块复用，并搭配 ELEMENT-UI 组件库的部分组件实现快速成型；同时使用 VUE-Router 作为前端路由响应页面请求，当需要动态获取数据并对页面进行局部更新时，前端通过 Ajax 方式访问后端 Restful 风格的接口，获取 JSON 格式的数据，并在前端完成解析和页面局部更新

应用接口层为后端和前端交互的枢纽，按照前端对页面的划分，后端使用 flask 中的蓝图技术将功能模块划分为用户类 (user)，文书类 (writ)，任务类 (task) 以及配置类 (config)，与前端保持一致，为前端提供符合 Restful 规范的接口。

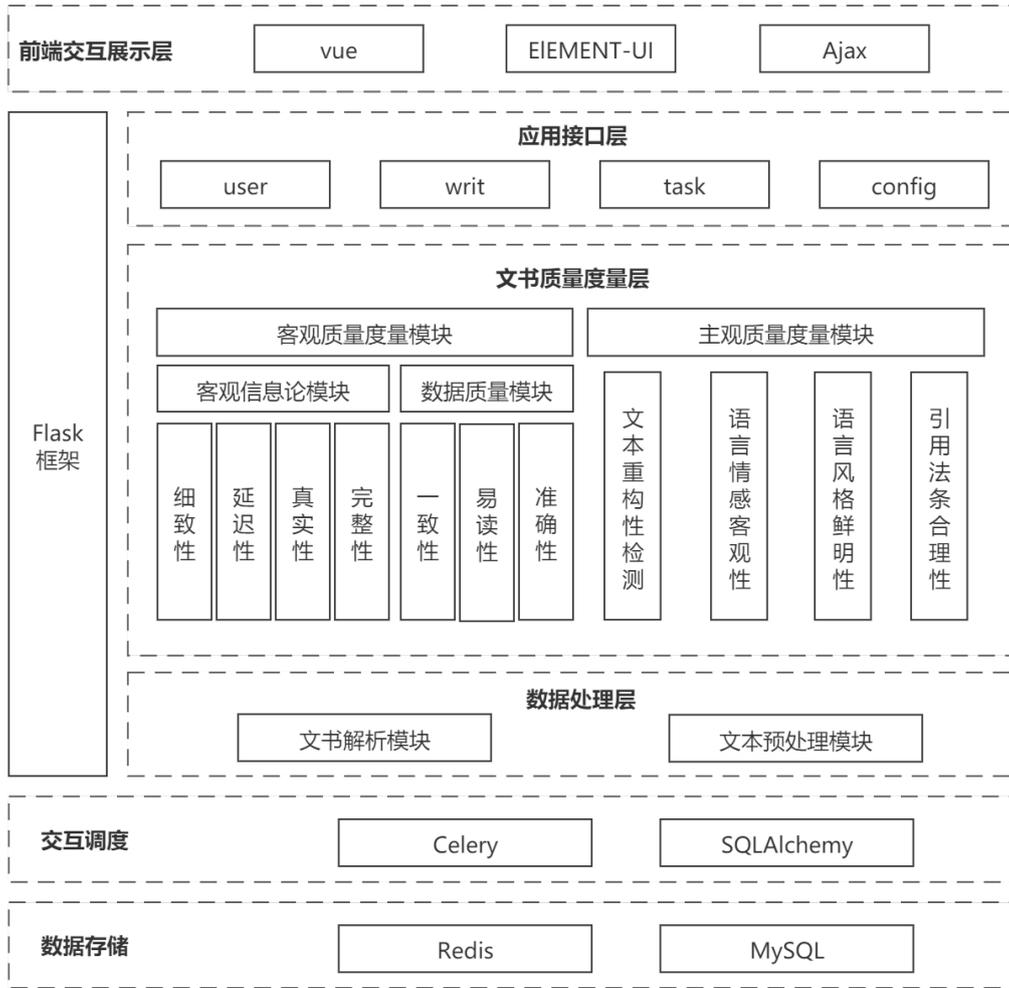


图 4.4: 系统整体架构设计图

文书质量度量层为本系统核心框架。结合前文提到的文书度量指标的设计，主要分为两个模块，第一部分是客观质量度量模块，对文书中的结构化标签信息质量进行度量，引用了客观信息论领域中的细致性、延迟性、真实性、完整性，以及数据质量领域的一致性、易读性、准确新，共计七个度量维度，其中每个质量度量维度又有若干个度量项，合计 15 个度量指标。第二部分是主观质量度量模块，对文书中的非结构化文本信息质量进行度量，本模块是建立在后台

存储的预训练模型及数据集的基础上,从文本重构性检测、语言情感客观性、语言风格鲜明性、引用法条合理性四个角度对文书进行度量。

数据处理层是连接数据存储层和文书质量度量层的纽带,主要分为文书解析模块以及文本预处理模块。系统接收文书数据格式为 xml 类型,所以需要根据标签及预置规则完成数据项的抽取,得到的结构化信息供客观信息质量度量模块进行度量,非结构化的文本信息则需要输入到文本预处理模块,进行分词或向量表示等预处理工作后,方可传输到主观信息质量度量模块完成度量。

交互调度层负责连接后端的应用服务操作和底层的数据存储。Celery 负责异步任务的调度,并且为系统后期的分布式扩展做准备,SQLAlchemy 负责数据对象到数据库中数据的映射,以及数据库连接和管理的操作。

数据存储层负责系统中全部数据的存储。文书数据和与训练的模型文件存储在文件系统中,为上层计算提供数据基础;MySQL 负责存储系统中用户配置信息和文书统计信息的存储和查询,保障用户管理功能的实现。Redis 负责存储 celery 异步任务队列及计算产生的结果信息。

4.3.2 4+1 视图

对于一个完整的系统来说,一个架构视图往往不能涵盖系统架构的全部内容,4+1 视图采用“分而治之”的思想,从特定的视角出发对系统进行简化,只描述该视角上系统的架构而省略与之无关的实体,最后对所有视图进行整合形成系统的完整架构。4+1 视图包括场景视图、逻辑视图、开发视图、过程视图及物理视图,分别对应了系统需求分析,概要设计、详细设计、系统开发以及最终部署的完整实现流程,由于前文中已经详细描述了用户需求,所以本小节省略场景视图,采用剩下的四个视图对本系统的整体架构进行介绍。

1. 逻辑视图

逻辑视图关注用户功能需求,其中不仅包含用户可见的功能,还要涵盖为用户提供功能所必须的中间模块,根据前文的用户功能需求分析对系统进行的模块划分,本系统的逻辑视图如下所示:

按照用户视角的前端页面进行逻辑划分,系统前端主要包括 *writ,task,config* 三类页面,后端使用蓝图技术对系统做同样的划分,不同的界面发起的请求由对应蓝图中的路由函数响应。同前端进行数据交互使需要使用到的格式转换等函数统一存储在 *web_utils* 包中,供所有路由函数使用。接收到请求后路由函数对调用 *service* 层的服务,系统主要提供的服务包括文书数据质量度量使用的 *computational_utils* 和统计分析使用的 *analysis_utils*,计算过程中产生的需要存储的对象需要调用 *model* 层的 *database* 包,由 ORM 负责将请求转化为数据库操

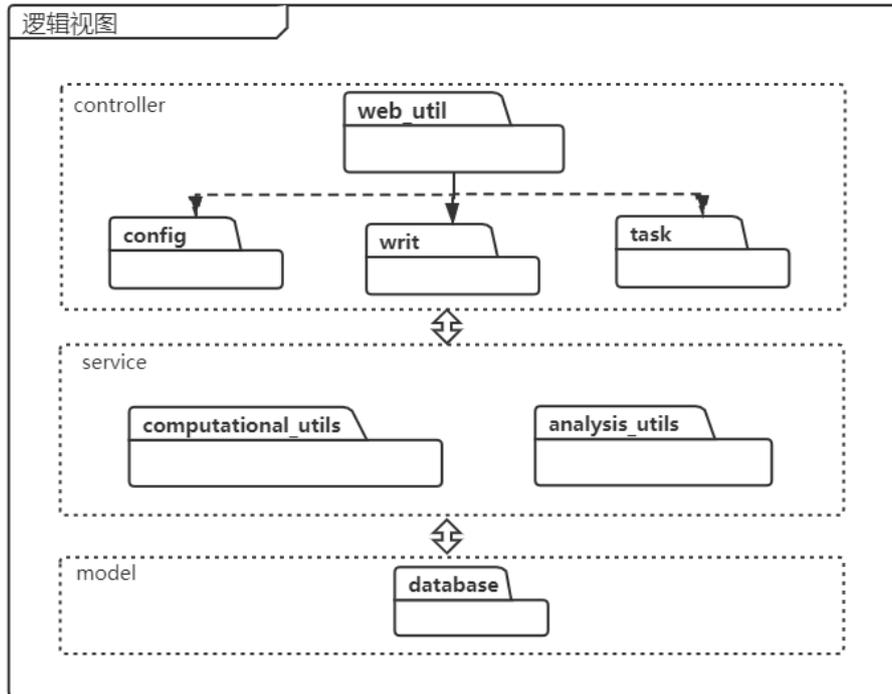


图 4.5: 文书质量评估系统逻辑视图

作语言，完成数据对象和数据库中记录的映射。

2. 开发视图

开发视图是系统程序包结构的静态体现，其中不但包括开发人员搭建的代码结构，还应该包含系统使用到的第三方框架、类库和系统运行时需要使用的软件及中间件，本平台的开发视图如图4.6所示：

系统后端开发基于 flask 框架完成，*web_parser* 负责前端请求的验证和参数解析，为前端和后端交互的中间人。对于响应用户请求的路由函数，系统使用 Blueprint 技术对其进行模块化处理，划分为 *my_config*、*writ* 以及 *task* 三个模块；*my_config* 负责处理用户关于配置方面的请求，包括修改和查询用户配置；*writ* 负责处理和文书相关的请求，包括上传文书，文书基本信息的查询以及文书质检等；*task* 负责平台中批量检测的文书任务管理，包括任务的创建，批量检测以及统计分析等。为前端提供标准 restful 规范接口，系统使用 flasgger 工具完成接口文档的编写和接口自动化测试，每个接口的文档以 yml 格式存储在 *api_description* 包中。*database* 负责与后端数据库进行交互，系统使用 SQLAlchemy 作为 ORM 工具，为系统提供结构化数据的存储和查询功能。

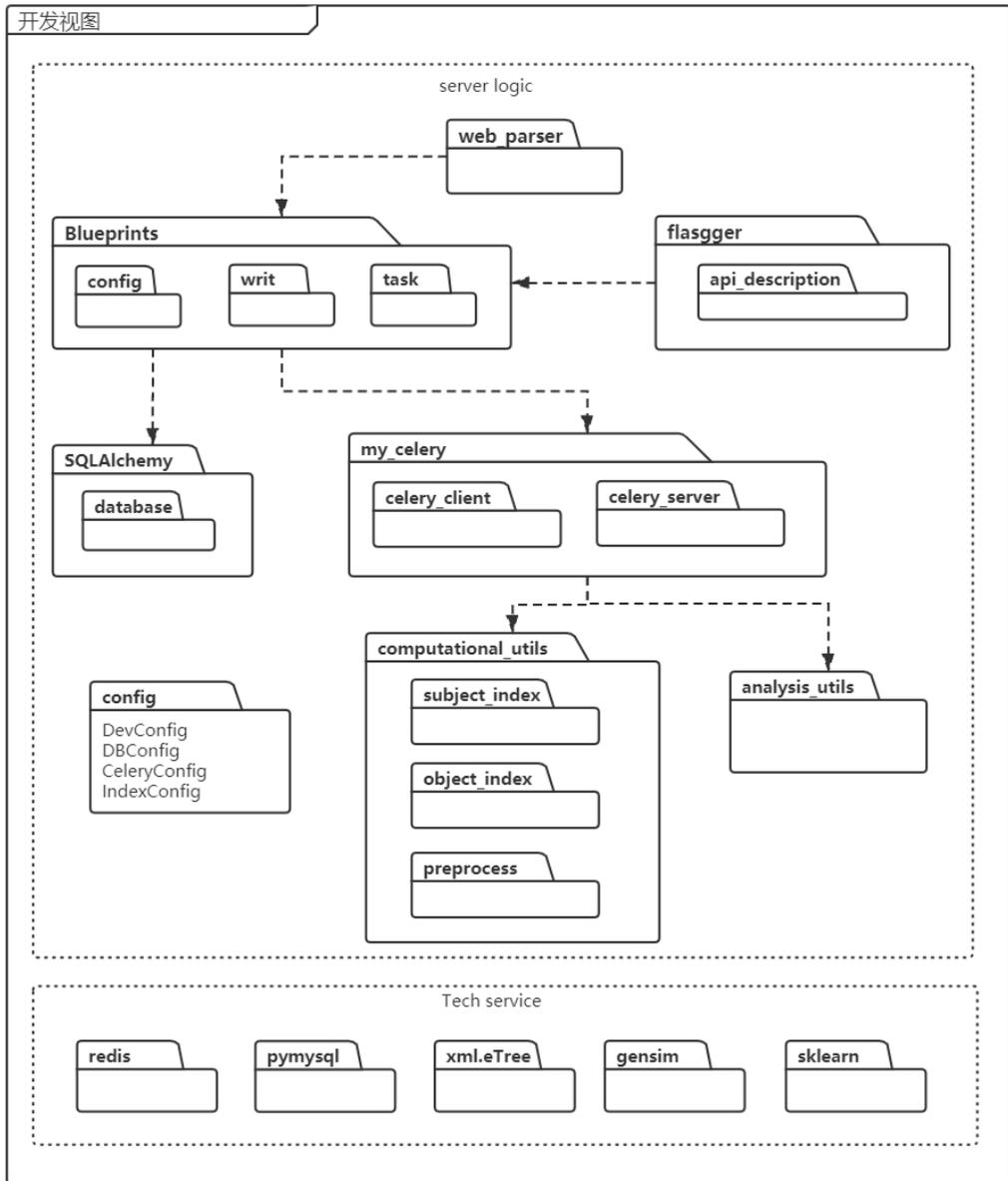


图 4.6: 文书质量评估系统开发视图

writ 与 task 发起的异步检测任务请求时需要经过 *my_celery*, 由 *celery_server* 进行任务调度, *my_celery* 会根据请求调用计算工具, 计算工具包括 *computational_utils* 及 *analysis_utils* 两部分, *computational_utils* 负责单篇文书的质量度量计算, *analysis_utils* 负责批量文书的统计分析计算。*computational_utils* 为质检系统的核心计算包, 其中包括主观语义质量度量计算 *subject_index*, 客观语法质量度量 *object_index* 及数据预处理 *preprocess* 三个模块。*config* 为后端各个模块提供配置信息, 其中包括 flask 框架需要的 *Devconfig*, 数据库配置信息 *DBConfig*、

Celery 配置信息 *CeleryConfig* 以及所有指标的配置 *IndexConfig*。为了实现上层逻辑系统，底层服务使用 redis 辅助 celery 和 redis 服务器的交互，pymysql 辅助 SQLAlchemy 与 mysql 数据库的交互，xml.eTree 完成文书的解析，gensim 和 sklearn 实现模型的训练和使用。

3. 进程视图

进程视图又叫处理视图，主要关注系统在运行过程中进程及线程之间的并发、通信、同步等交互关系，开发视图是程序的静态结构，那么进程视图就是程序运行时对象、线程、进程之间的交互过程，本平台的进程视图如下所示：

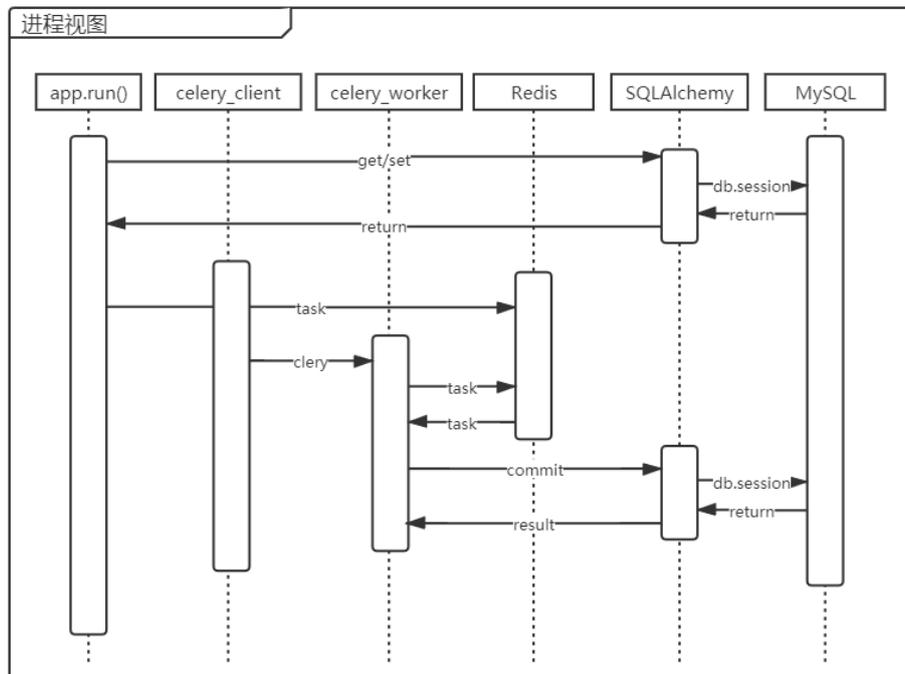


图 4.7: 文书质量评估系统进程视图

系统基于 flask 框架，在启动时会通过 *app.run()* 开启主线程响应前端请求。当前端发起系统管理相关的任务时，例如修改系统配置信息，文书信息，任务信息，后者获取信息等，系统会使用将对象的操作传给 SQLAlchemy，SQLAlchemy 的 core 将其转化为数据库语言后和数据库建立连接，进行对应操作，并获取数据库的操作结果返回给应用。当系统发起单篇文书质检或批量文书质检操作时，系统会将任务存储到 redis 消息队列中，以单篇文书质量度量为例，应用首先将 *writ_measure* 的任务发送给 redis，由 *celery_client* 创建的 *celery_worker* 会从消息队列中获取队伍进行度量运算运算，并将运算的结果通过 SQLAlchemy 写入数据库中，数据库会将写入的成功结果返回给 worker。

4. 物理视图

物理视图是将目标程序安装或部署到物理机器的依据，关注目标软件在使用过程中依赖的运行库和软件包，以及相互之间的通信问题，本平台的物理视图如下所示：

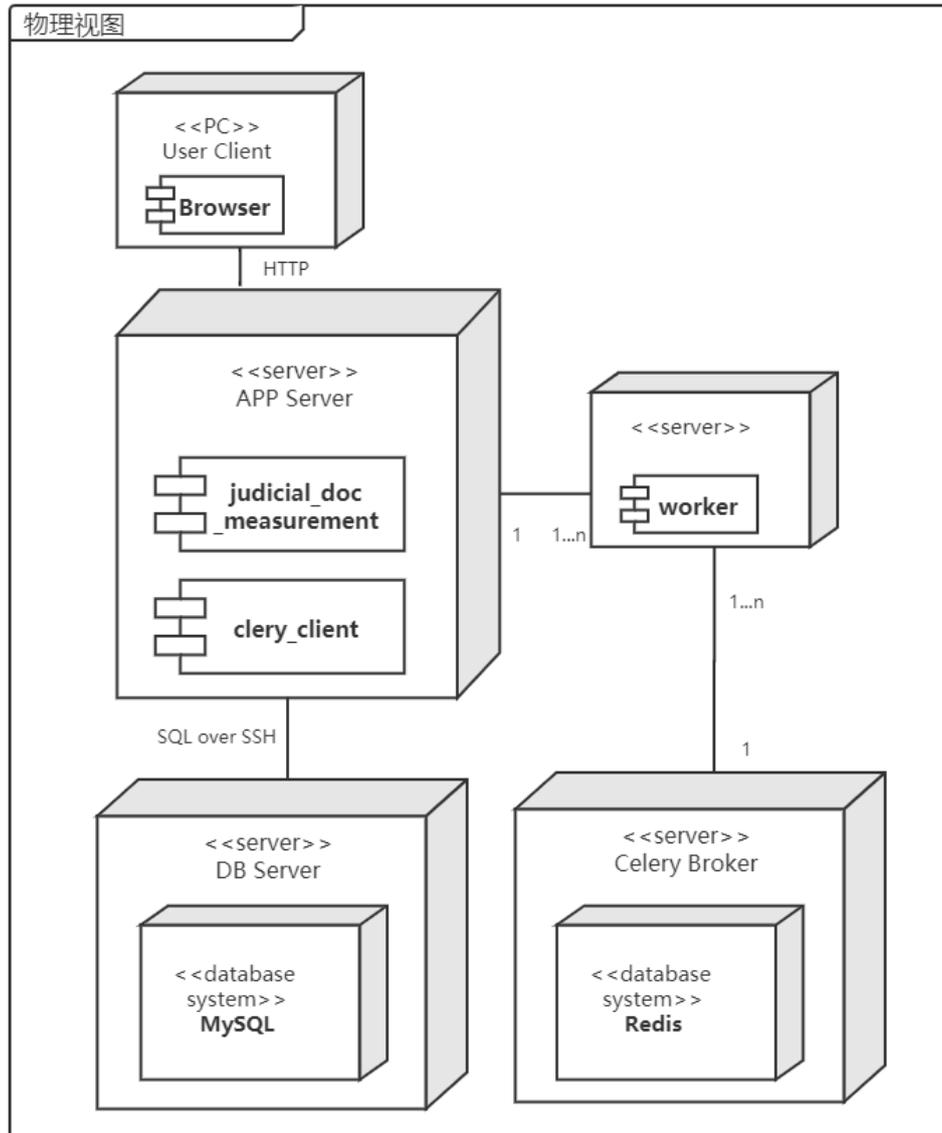


图 4.8: 文书质量评估系统物理视图

客户端和服务端之间通过 HTTP 协议进行数据交换，用户通过客户端的浏览器向服务端发起请求。服务端根据业务请求和底层数据存储分为两部分，上层负责请求响应及逻辑计算，底层负责文件、任务等资源的存储。*APPserver* 负责相应

前端发起的请求，主程序存储在 *judicial_doc_measurement* 组件中，*celery_client* 组件负责异步任务的调度，可以和主进程分开部署，分别启动。异步任务由 *celery* 中的 *worker* 执行，可以同时开启多个 *worker* 并行执行任务，前提是资源之间互不冲突，本系统中文书度量任务和分析任务之间相互独立，可以将任务分发给多 *worker* 同时作业。数据库服务器基于 MySQL 数据库系统实现，负责存储系统中使用的结构化数据，数据库服务器和 APP 服务器之间通过 SSH 协议沟通。异步任务队列使用 Redis 进行存储，*worker* 根据调度从 Redis 中获取任务进行执行。

4.4 数据交互模块设计

数据交互模块为平台对外提供的接口，对内提供文书数据来源，对外提供质检报告结果。数据交互模块的类图如图4.9所示，下面对模块中使用到的类进行详细介绍：*configs*

接受用户数据交互请求的类主要包括蓝图中的 *writ* 以及 *task* 类，*writ* 中 *upload_file* 为上传文件函数，接受前端传输的文件，并将传输成功的文件列表以 json 格式返回给前端。*writ* 中的 *get_writ_report* 及 *task* 中的 *get_task_report* 为获取报告函数，向前端传输度量结果文件的 json 格式。两个类中的 *db* 为 SQLAlchemy 的实例化对象，为系统初始化时根据 *DevConfig* 中 *SQLALchmey* 相关配置生成。

web_utils 为与前端交互所需调用的工具类，其中包括为满足时间数据格式需求的转换函数 *stamp2time* 及 *stamp2time*，以及对前端上传文件格式进行过滤的 *filter* 函数，对压缩文件进行解压处理的 *unzip* 函数。

Judicial 类、*Task* 类、*Report* 类、*Analysis* 类为数据交互模块涉及到的数据模型类，均继承自 *db.Model* 类，用户上传新的文件后，系统会为其创建 *JudicialDoc* 类的对象，对象中包括文书的基本信息 *id*、*user_id*、*task_id*、*docname*、*check_status*、*length*、*loc*、*province*、*writ_date*，属性定义同数据库定义中相同，所以在此不再详细叙述，*reports* 为该文书对应的质检报告，数据类型 *Report* 对象的列表，当添加外键为 *JudicialDoc* 对象的 *Report* 对象时，*reports* 中会自动更新列表。类似的，当用户创建新的任务时，系统会创建 *Task* 对象，其中除了 *Task* 的基本信息 *id*、*user_id*、*name*、*date*、*progress_status* 外，还包括其中包含的文书对象列表 *docs* 以及统计分析报告列表 *analysis_report*。

当用户下载文书报告时，系统使用 *db* 的 *query* 函数查找文书 *id* 对应的文书对象，获取 *reports* 属性，得到报告对象，通过对象的 *query()* 方法获取报告地址，得到报告返回给路由函数，路由函数将得到的报告传递给前端。获取分

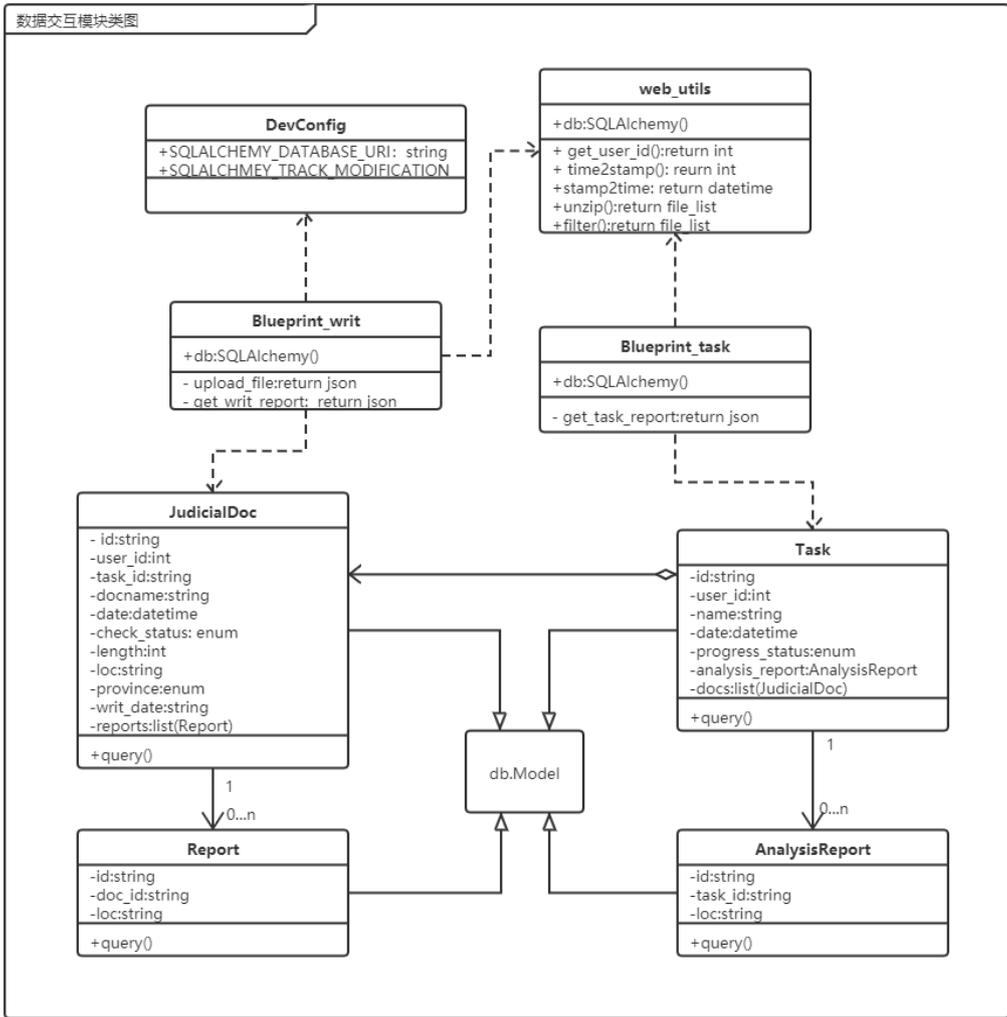


图 4.9: 数据交互模块设计类图

析报告的流程和文书报告相同，也是通过任务 id 获取任务对象，通过对象中的 *analysis_report* 属性获取分析报告对象，返回给路由函数。

4.5 用户数据管理模块设计

用户数据管理模块为平台的管家，负责平台所有数据的管理操作，该模块类图如图4.10所示。系统管理功能涉及的对象包括文书、任务、配置，系统蓝图按照相同逻辑划分，所以涉及的类包括 *writ*，*task* 及 *config*，用于响应前端对应的管理请求，*writ* 中包括 *get_writ_list()* 函数，获取当前用户权限下的文书列表，返回为列表信息的 json 格式。*task* 中包括 *get_task_list()* 函数，负责获取当前用户创建的任务列表，返回为任务列表信息的 json 格式。*config* 中包含获取当前

用户配置的 *get_config()* 函数及修改用户配置函数 *set_config()* 函数，分别返回配置的 json 格式文件和是否修改成功的 bool 值。*DevConfig* 为数据库提供基本配置信息。

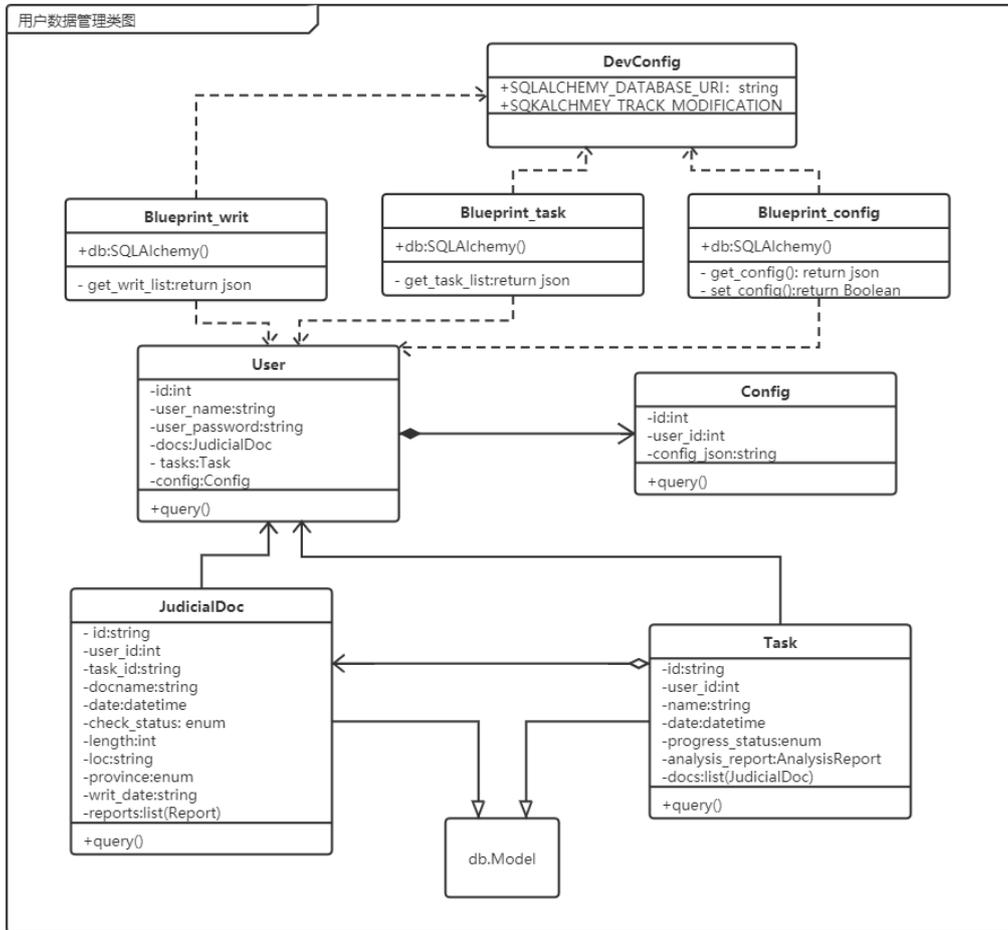


图 4.10: 用户数据管理模块设计类图

用户数据管理涉及底层 *Models* 中的 *User*、*Config*、*JudicialDoc* 及 *Task* 类，四个类均继承自 *db.Models*，当前用户登录时，会根据用户 *id* 获得用户对象，对象中包含当前用户的 *id*、*user_id*、*user_password* 基本信息，说明见数据库设计部分 *User* 表的字段说明。除此之外，属性中还包括用户权限下的文书列表 *docs*，列表中对象类型为 *JudicialDoc*，任务列表 *tasks*，列表中对象类型为 *Task*，以及用户对应的配置文件 *config*，类型为 *Config*，用户存在才有对应的配置信息，所以 *Config* 对象与 *User* 对象之间为组合关系。通过 *User* 的 *docs*、*tasks* 及 *config* 三个属性即可获得当前用户所有的用户管理数据对象，用户在对象的基础上即可完成数据的修改和查询操作。

4.6 质量检测模块设计

质量检测模块为本系统的核心模块，为文书提供全面的度量检测服务，度量方法涵盖客观信息质量度量、客观数据质量度量及主观数据质量度量。该模块的类图如4.11所示：

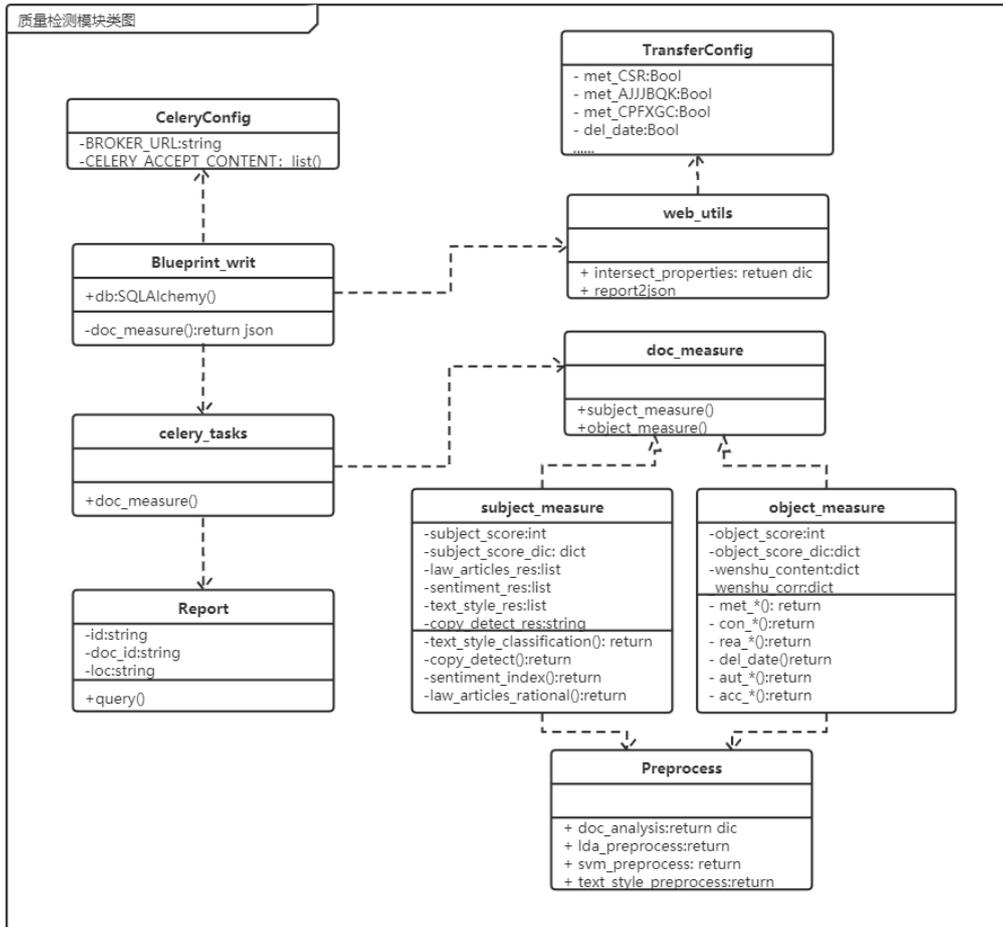


图 4.11: 文书质量检测模块设计类图

Blueprint_writ 中的 *doc_measure()* 负责响应前端发起的文书质量检测请求，接收参数包括待检测的文书 id 以及度量使用的指标字典。度量指标可以由用户在发起检测任务时手动设定，也可选择默认配置，这要借助 *web_utils* 中提供的 *intersect_properties()* 函数进行处理，结合 *TransferConfig* 中标准度量指标得到最终的检测指标字典。由于文书度量时间长短不一，为减少用户等待时间，系统自动将任务交给后台异步处理。*celery_tasks* 中声明了系统中需要异步完成的 celery 任务，celery 会将接收到的异步任务存储至消息队列中，由 worker 从队列

中获取任务执行。

subject_measure 和 *object_measure* 为度量模块的核心类,两者均实现 *doc_measure* 接口。*subject_measure* 属性包括主观质量度量总分 *subject_score*, 各分项得分字典 *subject_score_dic*, 相关法条推荐列表 *law_articles_res*, 案件描述段落分句情感得分字典 *sentiment_res*, 案件描述分句文本风格鲜明性得分字典 *text_style_res* 以及疑似抄袭文本 *copy_detect_res*。度量方法包括前文指标设计中提到的文本风格鲜明性度量 *text_style_classification()*, 文本抄袭检测度量 *copy_detect()*, 文本情感客观性度量 *sentiment_index()* 以及引用法条合理性度量 *law_articles_rational()*。

object_measure 属性包含客观度量总得分 *object_score*, 各指标分项得分字典 *object_score*, 文书篇章结构分段内容字典 *wenshu_content* 以及每个分段对应的质量修改意见字典 *wenshu_corr*。度量方法和上文客观质量度量指标设计相对应, *met_** 为细致性度量方法, 包括参诉人信息细致性 *met_CSR*、事实部分细致性 *met_AJJBQK*, 理由部分细致性 *met_CPFXGC*; *con_** 为一致性度量方法, 包括数字使用一致性 *con_num* 及标点符号使用一致性 *con_pun*; *rea_** 为易读性度量方法, 包括事实描述部分简明性 *rea_SSMS*, 争议焦点条理性 *rea_ZYJD*; *acc_** 为准确性度量方法, 包括构成事项准确性 *acc_GCSX*、审理经过准确性 *acc_SLJG*, 参诉人信息准确性 *acc_CSR*; *aut_** 为真实性度量, 包括俺有信息类别规范性 *aut_AY* 及裁判依据引用规范行 *aut_CPYJ*; *del_date* 为延迟性度量。

celery worker 会根据调度从消息队列中获取检测任务, 然后根据指标配置文件选择调用的度量方法, 为了加强系统的可扩展性, 方便后续度量指标的加入, 系统在设计时已经对各个度量方法的返回值做了统一, 汇总得到的结果会借助 *web_utils* 类中的 *report2json* 函数转换成前端展示需要的 *json* 格式存储, 同时新建 *Report* 对象, 对象属性包括 *id*, 对应文书标识 *doc_id* 以及文书存储的地址, 更新数据库中信息。

Preprocess 类为度量方法提供数据的预处理服务, 其中包括为客观度量提供的原始文书解析函数 *doc_analysis*, 为主管度量提供文本预处理的 *lda_preprocess*、*svm_preprocess* 以及 *text_style_preprocess* 函数。

4.7 统计分析模块设计

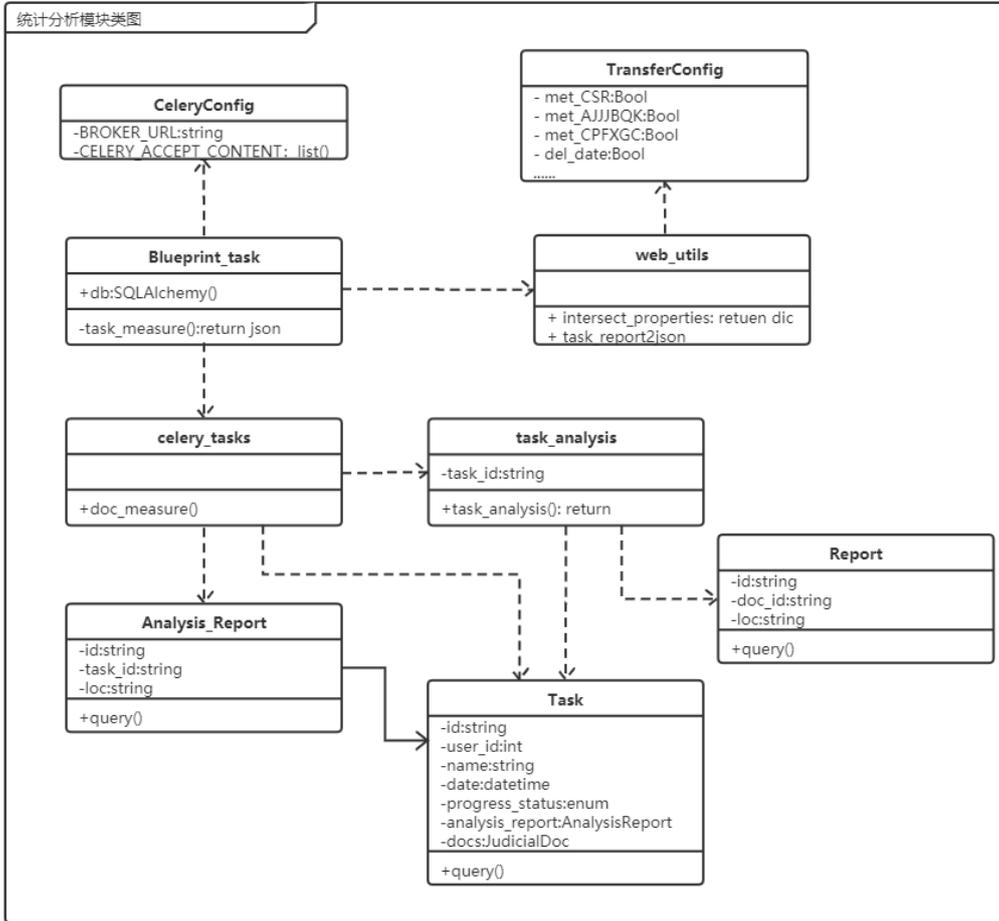


图 4.12: 统计分析模块设计类图

统计分析功能是在文书度量完成的基础上实现，该模块的类图如4.12所示，响应统计分析请求的为 *task* 蓝图中的 *task_measure()* 函数，请求参数包括任务 *id* 和度量配置字典，首先调用 *web_utils* 中的 *intersect_properties()*，结合定义好的 *Transfer_config* 得到标准配置字典，然后将统计分析任务交给 *celery* 执行，*celery* 在获取任务后会调用 *task_analysis* 对象中的 *task_analysis()* 方法完成统计分析任务。

task_analysis() 首先会根据属性 *task_id* 获取对应的 *Task* 对象，然后通过 *Task* 对象中的 *docs* 属性获取 *JudicialDoc* 对象列表，文书对象中包含统计需要用到的省份 *province*，文书时间 *writ_date* 信息，以及对应的 *report* 对象，从 *report* 对象中获取文书报告的存储位置，得到蜗牛属的度量结果信息。分析过程

会对任务的文书列表中的所有文书进行如上操作，并将得到的信息进行汇总分析，最后使用 *web_utils* 中提供的 *task_report2json()* 函数得到分析报告的 json 格式返回给前端。同时创建 *AnalysisReport* 对象，对象属性包括 *id*、*task_id*、*loc*，并更新 *Task* 中的 *progress_status*。

4.8 数据库设计

4.8.1 ER 图

本小节主要介绍系统中使用的数据库结构。文书作为 xml 文件，主要以文档的形式存储在服务器中，但是文书和用户信息等需要频繁查询更改的信息存储在关系型数据库中，主要涉及到的实体和实体之间的关系如下图所示。

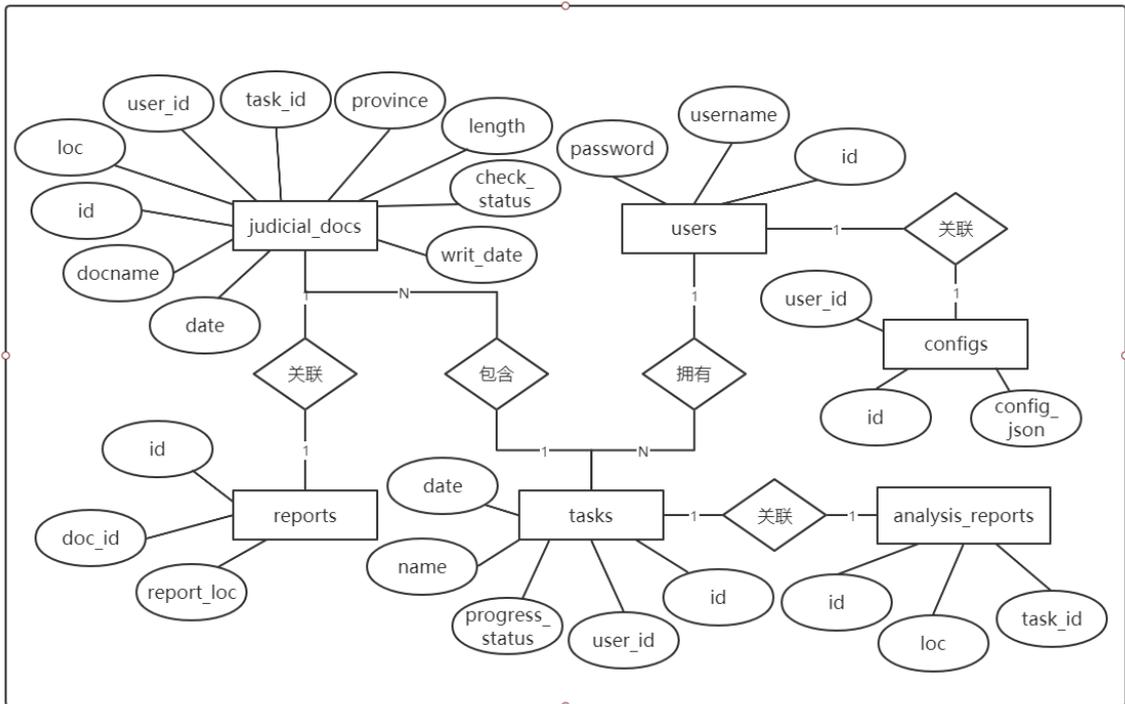


图 4.13: ER 图

数据库中主要包含的表结构有 *judicial_docs*、*users*、*reports*、*tasks*、*analysis_reports* 以及 *configs*。设计围绕文书展开，文书属性不但包括文书上传时间，文书长短等作为系统对象的属性，也包括解析后得到的文书写作日期和地点等内

容属性。用户和文书之间为一对多的关系，即用户可以上传多篇文书，但是即便内容重复，一篇文书也只能属于一个用户，保证用户之间信息的独立性和安全性。*judicial_docs* 和 *reports* 相关联，*reports* 为文书的质检报告相关信息，一篇文书只能对应一篇质检报告，一经生成不可更改，系统会将其以 json 文件的格式存储在服务器中，并将存储地址保存在数据库中。*tasks* 为用户自定义的任务，一个用户可以创建多个任务，而 *tasks* 和 *judicial_docs* 之间为一对多的包含关系，用户可以选择为多篇文书创建一个批量检测的任务，方便系统对大规模数据进行离线检测，数据库中记录了任务的创建时间和检测状态等信息。和单篇质检报告类似，对于批量检测的任务，系统会为其数据集生成统计分析报告，相关的信息存储在 *analysis_reports* 表中。除此之外，系统为用户提供自定义检测项的定制化功能，所以数据库中会将用户默认的配置信息存储在 *configs* 表中。*configs* 表和 *users* 为一一对应的关系，即一个用户只能有一套配置。

4.8.2 数据库表设计

1. 文书表 (*judicial_docs*): 文书表存储了文书相关的所有结构化信息，包括系统信息和内容信息两部分，上传文书后数据库会存储文书唯一标识 *id*，文书在服务器中的存储地址 *loc*，文书上传时间 *date*，上传时的文件名称 *docname*，上传文书的用户标识 *user_id*，文书文件大小 *length* 以及文书的检测状态 *check_status*，初始化为 *untested*，在文书质检后数据库中会更新文书的内容信息，包括文书所属省份 *province*，文书所属年份 *writ_date*，并更新对应的 *check_status*。

表 4.10: *judicial_docs* 表

字段	类型	允许为空	说明
<i>id</i>	<i>varchar(50)</i>	False	文书唯一标识
<i>user_id</i>	<i>int</i>	False	上传文书的用户 <i>id</i>
<i>task_id</i>	<i>int</i>	True	文书属于的任务 <i>id</i>
<i>docname</i>	<i>varchar(50)</i>	False	用户上传时的文件名称
<i>date</i>	<i>datetime</i>	False	用户上传文书时间
<i>check_status</i>	<i>enum</i>	False	文书当前状态
<i>length</i>	<i>int</i>	False	文书文件大小
<i>loc</i>	<i>varchar(200)</i>	False	存储地址
<i>province</i>	<i>varchar(10)</i>	True	隶属省份
<i>writ_date</i>	<i>varchar</i>	True	文书时间 (年份)

2. 报告表 (*reports*): *reports* 主要存储单篇文书的结构化信息。包括报告的唯一标识 *id*，报告对应的文书标识 *doc_id*，报告在服务器中的存储地址 *report_loc*。

表 4.11: reports 表

字段	类型	允许为空	说明
id	varchar(50)	False	报告唯一标识
doc_id	varchar(50)	False	报告对应的文书 id
report_loc	varchar(200)	False	报告存储位置

3. 用户表 (*users*): 用户表主要记录登录系统的用户信息, 为了方便后续和公司用户系统进行对接, 只包含必须的用户名称和密码字段, 包括用户的唯一标识 *id* 以及用户登录需要的用户名 *username* 和密码 *password*。

表 4.12: users 表

字段	类型	允许为空	说明
id	int	False	用户唯一标识
username	varchar(50)	False	用户名称
password	varchar(50)	False	用户密码

4. 任务表 (*tasks*): 任务表是对文书的抽象, 由用户选中多篇文书后创建。其属性包含任务的唯一标识 *id*, 用户指定的任务名称 *name*, 用户创建任务的时间 *date*, 建立任务的用户标识 *user_id*, 以及任务内报告是否检测完成并生成分析报告的 *progress_status*。

表 4.13: tasks 表

字段	类型	允许为空	说明
id	int	False	任务唯一标识
user_id	int	False	创建任务的用户 id
progress_status	enum	False	任务当前状态
name	varchar(50)	False	用户创建时的任务名称
date	datetime	False	用户创建任务时间

5. 分析报告表 (*analysis_reports*): 和 *reports* 表类似, *analysis_reports* 负责存储 *task* 对应的统计分析报告表的信息, 属性包括唯一标识 *id*, 对应的 *task_id*, 统计分析报告存储的位置 *loc*。

表 4.14: configs 表

字段	类型	允许为空	说明
id	int	False	配置唯一标识
user_id	int	False	对应用户 id
config_json	varchar(100)	False	json 格式配置信息

表 4.15: *analysis_reports* 表

字段	类型	允许为空	说明
id	varchar(50)	False	分析报告唯一标识
task_id	int	False	分析报告对应任务 id
loc	varchar(200)	False	分析报告存储位置

6. 配置表 (*configs*): *configs* 记录每个用户默认的质量检测指标配置, 属性包含唯一标识 *id*, 对应的用户标识 *user_id*, 定义好的 json 格式的配置信息 *config_json*。

4.9 本章小结

本章首先定义了系统边界, 明确了系统在司法基本服务库中的角色和总体功能划分, 在此基础上展开系统的需求分析和设计。需求分析主要分为用户数据交互管理、文书质量检测与分析两部分, 以用例图和用例描述表的形式对详细的功能模块及功能点进行分析。系统设计包括系统概要设计和详细设计两部分, 概要设计使用 4+1 视图从多个角度对系统整体进行介绍, 然后以类图的形式对每个模块的详细设计展开描述, 最后对系统中使用的数据库结构进行描述。

第五章 民事裁判文书质量评估系统实现

5.1 数据交互模块实现

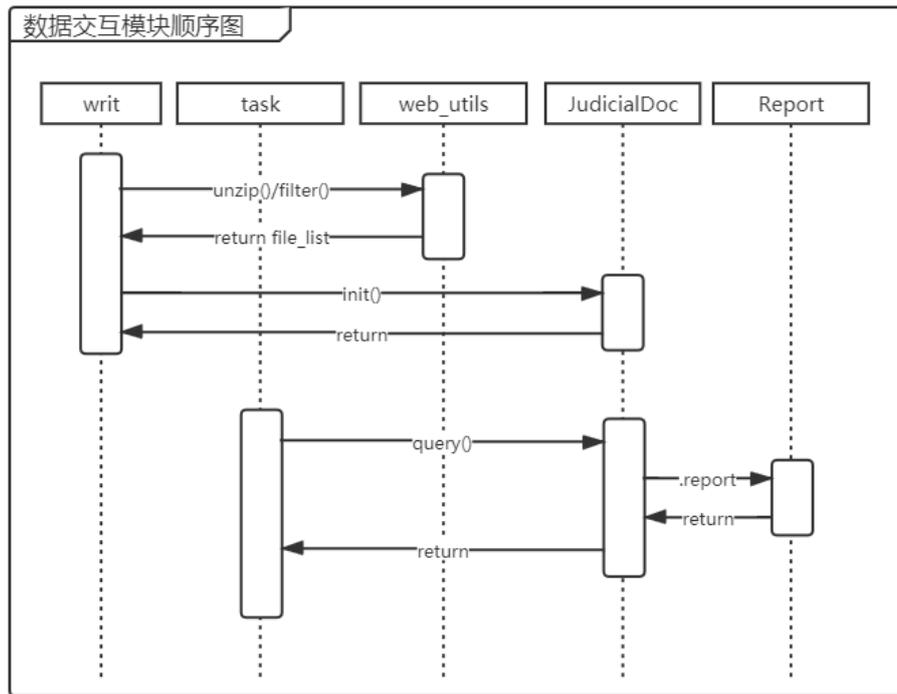


图 5.1: 数据交互模块顺序图

图5.1为系统数据交互模块中函数调用的顺序图，系统收到用户的上传请求后首先调用 *web_utils* 中的 *filter()* 和 *unzip()* 函数对上传文件进行过滤和解压操作，然后将过滤后的文件列表返还给 *writ*，*writ* 调用 *JudicialDoc* 创建文书对象，最后使用 *db.session* 将数据对象映射为数据库中实例化数据进行存储，并将存储的结果返回给 *writ*，*writ* 将存储成功的文件列表返回给前端。当用户发起下载请求时，*task* 中的路由函数根据请求中的文书 id 查找对应的 *JudicialDoc* 对象，通过文书对象中的 *reports* 属性获取 *report* 对象，最后将 *report* 中的地址属性 *loc* 中的值返回给路由函数，*task* 根据返回的文书地址读取文件得到报告结果，将其封装成接口规定格式后传给前端。由于对单篇文书结果下载和统计报告下载流程相似，所以流程图中只描述了单篇文书的下载。

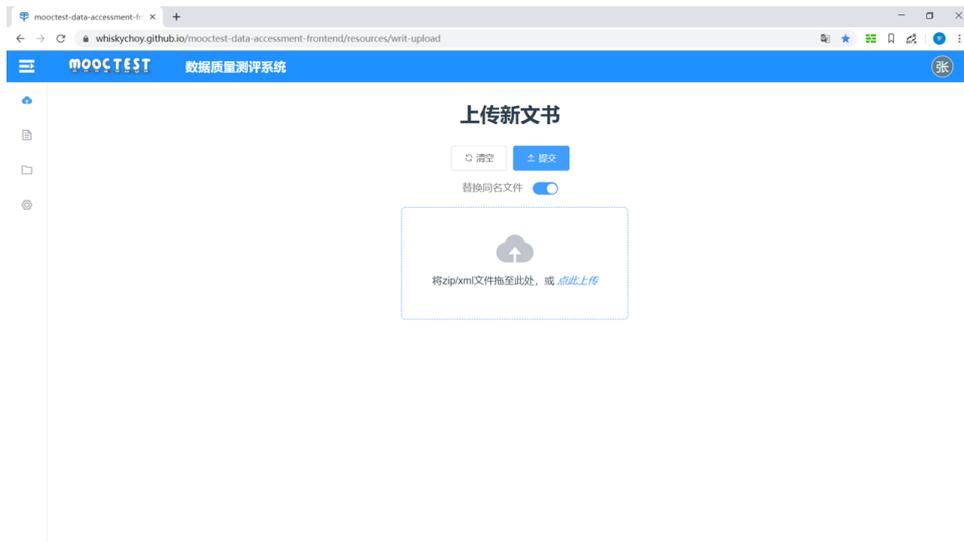


图 5.2: 系统文书上传页面

文书上传的界面如图5.2所示，用户在我的文书列表中点击上传文书即可进入文书上传页面，点击选择文书即可浏览本地文件目录选择文件上传，用户可以一次选择多个文件上传，也可将文件打包成压缩包统一上传，交由后端完成解析，解析后得到的文书名称中会包含压缩包的原始名称及具体文件名称，方便用户检索或分类。

5.2 用户数据管理模块实现

用户数据管理模块主要分为文书、任务及配置管理，涉及页面如下所示：

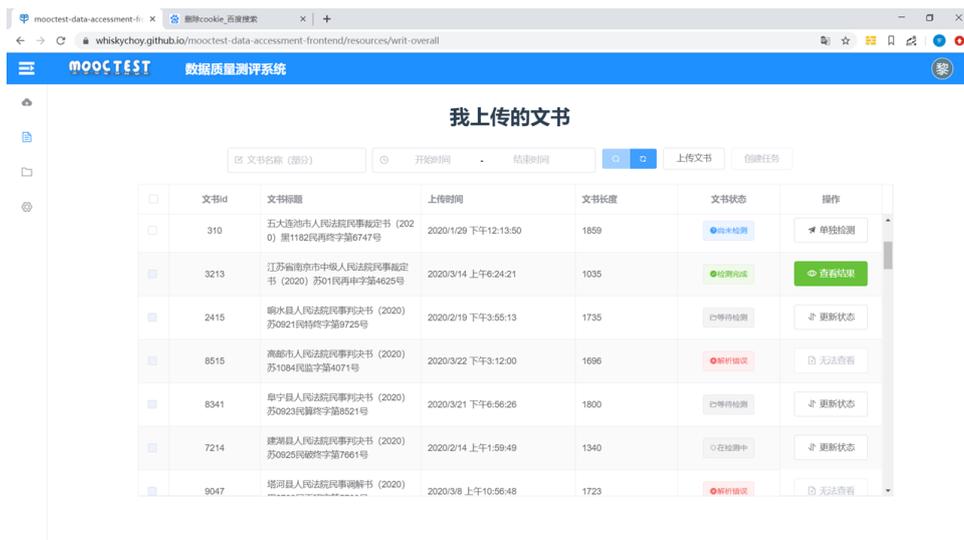


图 5.3: 系统文书管理页面

第五章 民事裁判文书质量评估系统实现

文书管理功能对应我的文书页面5.23，用户可以在列表中查看所有已上传的文书基本信息、当前状态以及可执行的操作，并且支持用户根据关键字段对文书进行检索，帮助用户快速定位，例如上传压缩包文件时，系统会在文书名中加入压缩包名称作为前缀，用户搜索包名即可获取包内所有文书。基本信息包括文书的唯一标识 id、文书 id，上传时间及文书状态，文书状态主要分为五种，分别为尚未检测、等待检测、在检测中、检测完成和解析错误，用户可以点击更新状态刷新单个任务的状态，也可以刷新全页面获取所有任务最新信息。对于检测完成的文书，用户可以选择查看结果详情报告。

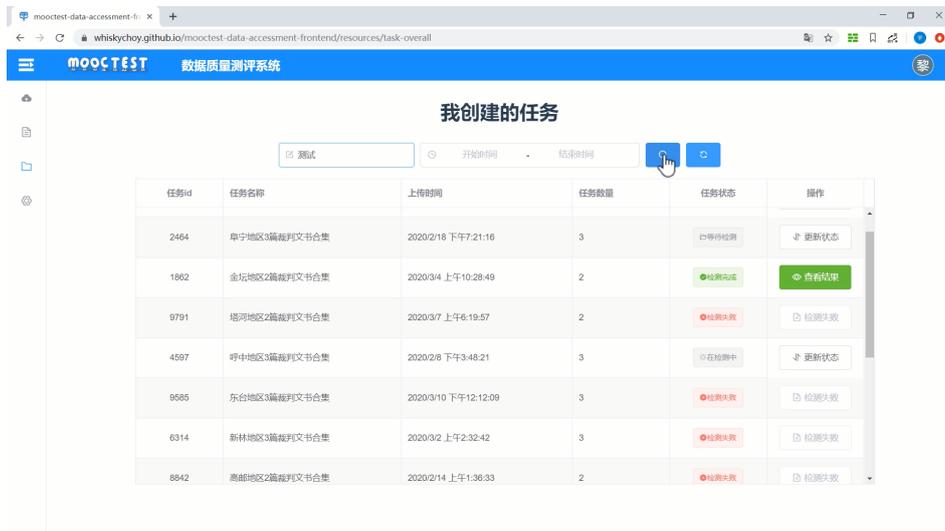


图 5.4: 系统任务管理页面

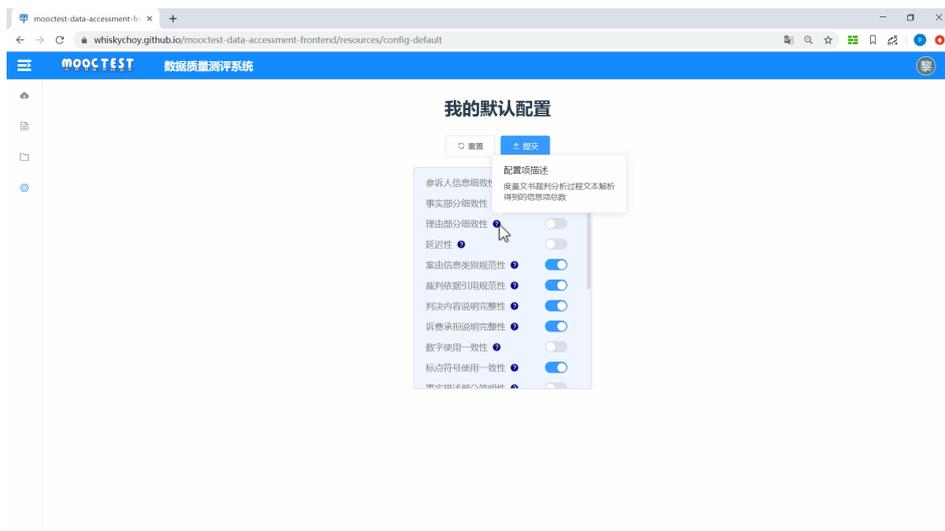


图 5.5: 系统配置管理页面

对于批量检测任务，用户首先需要在我的文书列表中选择需要检测的文书，创建新的检测任务，用户点选时右上角会显示当前已选中的文书个数，创建任务时用户可以自定义任务名称及检测时需要用到的度量指标。创建成功的任务可以在我的任务中查看，页面展示了任务的基本信息及任务当前状态和可执行的操作，同我的文书的操作相类似，因此不再详细叙述。

配置管理对应我的配置页面5.5，方便用户修改默认配置，每个配置项的简要度量方法都会在此简单介绍，方便用户根据需求进行选择，提交默认配置后，用户可以在每次执行质检任务之前选择默认配置。

5.3 文书质量检测模块实现

5.3.1 客观度量指标的实现

系统实现根据上文定义的度量指标进行实现，实现过程中指标和文书标签的对应关系如表所示，除了对应到文书中具体段落的质量度量，还由一些指标覆盖范围为全文，包括案件信息延迟性，裁判文书延迟性、构成事项准确性及数字使用一致性。

表 5.1: 度量指标及文书篇章结构对应关系表

事项结构	事项名称	xml 结构对应标签	度量指标
标题	标题	文首 (WS)	无
正文	首部	当事人 (DSR) 诉讼记录 (SSJL)	参诉人信息准确性 参诉人信息细致性 审理经过准确性 案由信息类别规范性
	事实	案件基本情况 (AJJBQK)	事实部分细致性 事实描述部分简明性
	理由	裁判分析过程 (CPFSGC)	理由部分细致性 争议焦点条理性
	裁判依据	法律法条引用 (FLFTYY)	裁判依据引用规范性
	裁判主文	判决结果 (PJJG)	判决内容说明完整性
落款	尾部	诉讼费承担 (SSFCD) 等	诉费承担说明完整性
落款	落款	文尾 (WW)	无
全文	全文	全文 (QW)	案件信息延迟性 裁判文书延迟性 构成事项准确性 数字使用一致性

论文中使用的数据为解析软件解析后得到的民事一审裁判文书，格式为 xml 格式，共计 17w 余篇，对标题进行规则匹配过滤后得到判决书 762122 篇。随机

抽取 50000 篇后组成客观信息质量度量部分的原始数据集，抽样检测发现其中包含解析结果错误造成的标签内容错误，例如案件基本情况中只包含查明事实段内容，为避免标签内容对后续的裁判文书评估造成影响，对文书的标题和案件基本情况等关键节点信息进行规则过滤，得到文书共 45768 篇。

对数据集中的文书进行地域分布统计和时间分布统计，得到的统计结果如下图所示。图5.6为裁判文书地域分布的统计图，其中裁判文书数量较多的省份包括浙江省、山东省、江苏省，较少的地区有新疆维吾尔自治区、海南省、青海省。图5.7是文书时间分布的统计图，按照年份进行划分，可以看出数据主要集中在 2013 年至 2015 年。

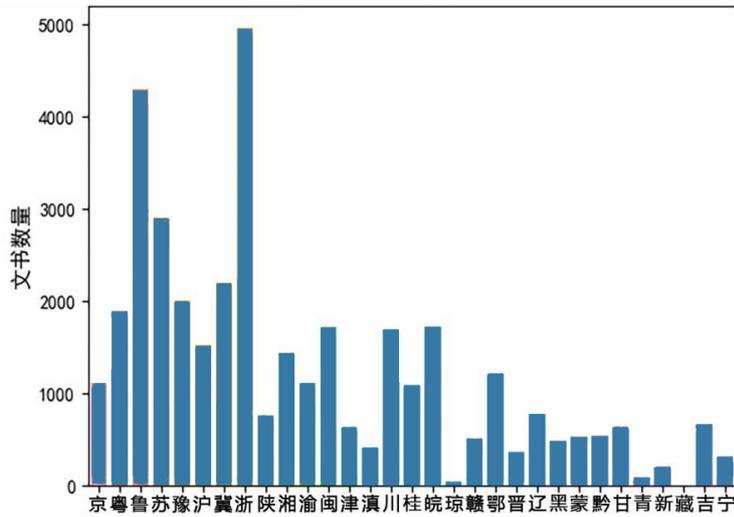


图 5.6: 文书数据集地域分布频数图

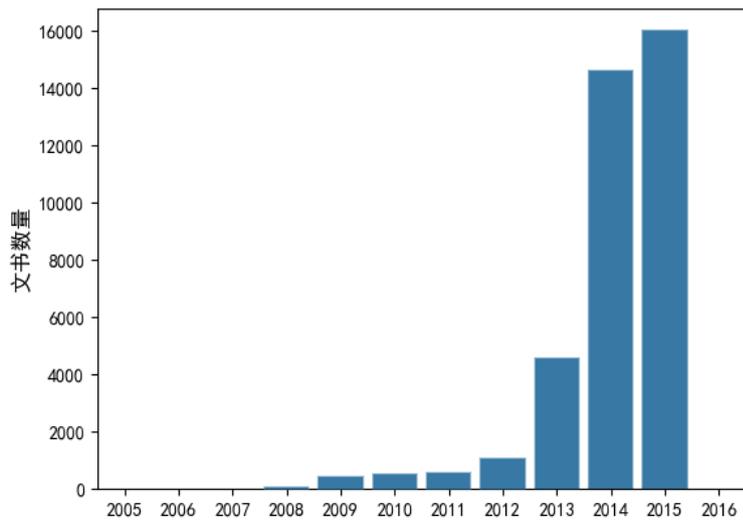


图 5.7: 文书数据集时间分布频数图

1. 细致性阈值设定

文书细致性度量指标分为参诉人信息细致性，事实部分细致性以及理由部分细致性，实验对数据集中 4w 余篇裁判文书的三种细致性信息进行统计，得到的琴图如图 5.8 所示。

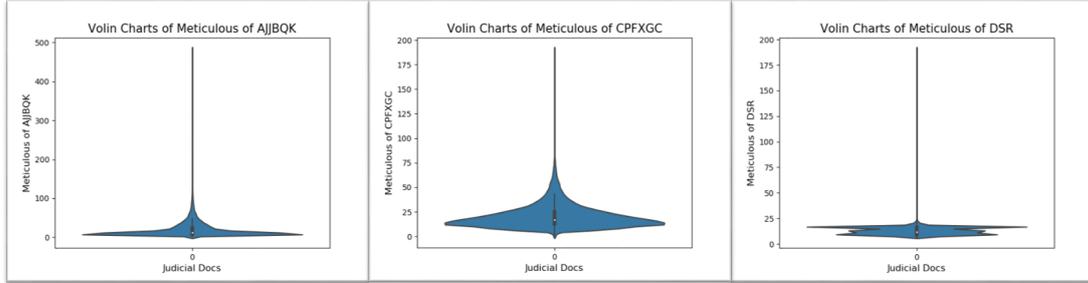


图 5.8: 细致性度量值数据统计 violin 图

为去掉个别过长案例对整体分布造成的影响，首先过滤掉排序后的后百分之五的数据，得到的三个指标值的密度图如 5.9 所示。

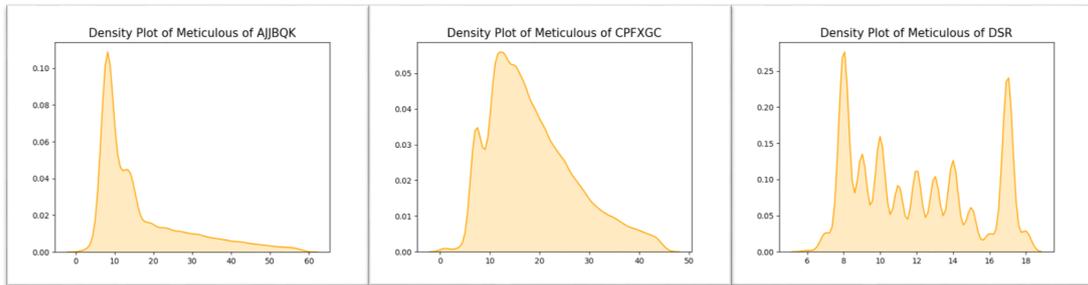


图 5.9: 细致性度量值数据统计 violin 图

由图可得，事实部分细致性和理由部分大体满足正态分布，根据置信度的计算公式，我们取 $\mu - 1 * \sigma$ 作为置信区间的下区间，结合细致性的实际意义，对置信区间的上区间不做要求，最终得到案件基本情况的阈值为 4.928，取整为 5；裁判分析过程的阈值为 8.39，向下取整为 8。参诉人信息细致性的图像存在多个峰值，这是由于不同类型的案件参诉人需要描述的信息项不同，例如法律工作者会涉及到工作单位，职位，公司等信息，而自然人信息项则相对较少，所以我们取最小峰值作为参考，将参诉人信息的细致性的阈值设置为 7。

2. 延迟性阈值设定

按前文定义延迟性分为案件信息延迟性集裁判文书延迟性。案件受理时间，案件结案时间可通过标签对文书进行抽取获得，案件发生时间通过筛选文书中

所有出现时间的最早时间获得，由于文书书写不规范或标签识别错误导致的时间获取值为空情况不在统计范围内。对数据集中裁判文书进行数据统计后共获得案件信息延迟数据 24628 个，文书信息延迟数据 26698 个，数据的单位均为天，两种数据的统计分布密度图如下所示。

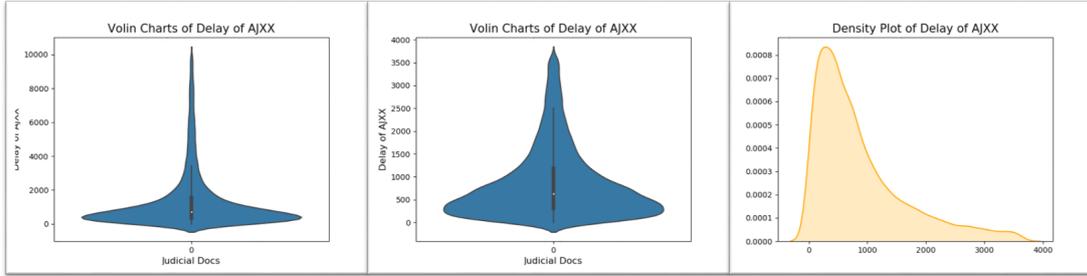


图 5.10: 案件信息延迟性统计分布图

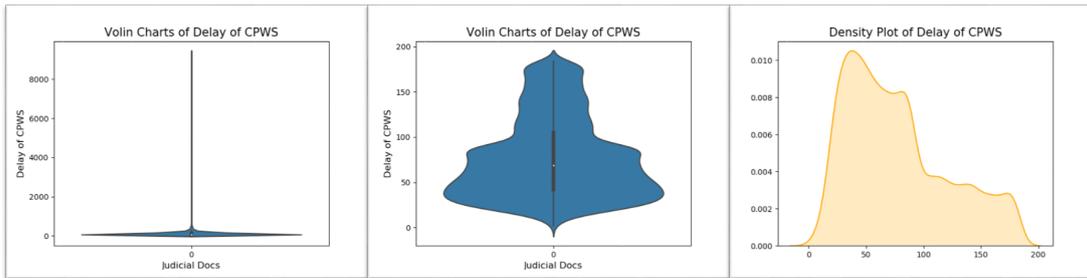


图 5.11: 裁判文书延迟性统计分布图

经过筛选后案件信息的延迟时间呈长尾分布，但大多集中在 1000 天以内，3000 以后数据极少，所以我们将案件信息的延迟性划分为三个等级，中间阈值为 1000 和 3000。裁判文书的信息延迟时间平均为 99 天，图中也可看出文书大多数数据分布在 100 以内，100-200 之间分布平稳，所以我们将裁判文书的延迟性划分为 3 个等级，100 天以内，100-200 天及 200 天以上。

3. 简明性阈值设定

根据对裁判文书细致性的度量方法，实验分别对数据集中裁判文书的原告诉称段、被告辩称段、查明事实段及证据段四段文本的长度进行了统计，得到四类数据的数量分别为 45426、44011、45217、49820，初步统计四类文本的长度 violin 图如 5.12 所示，过滤掉个别特殊案例后得到其各自的密度分布图如 5.13 所示。由图可见，原告诉称段及查明事实段的长度基本符合正态分布，同样取 $\mu - 1 * \sigma$ 作为置信区间的下区间，得到原告诉称段的阈值为 381.49，向下取整得到 381，查明事实段的阈值为 612.9，向上取整得到 613。被告辩称段及证据段

的长度为长尾分布，大部分数值集中在 50 字以内，如“被告 xxx 未作答辩”，设定被告辩称段的长度阈值为 187，证据段的长度阈值为 200。

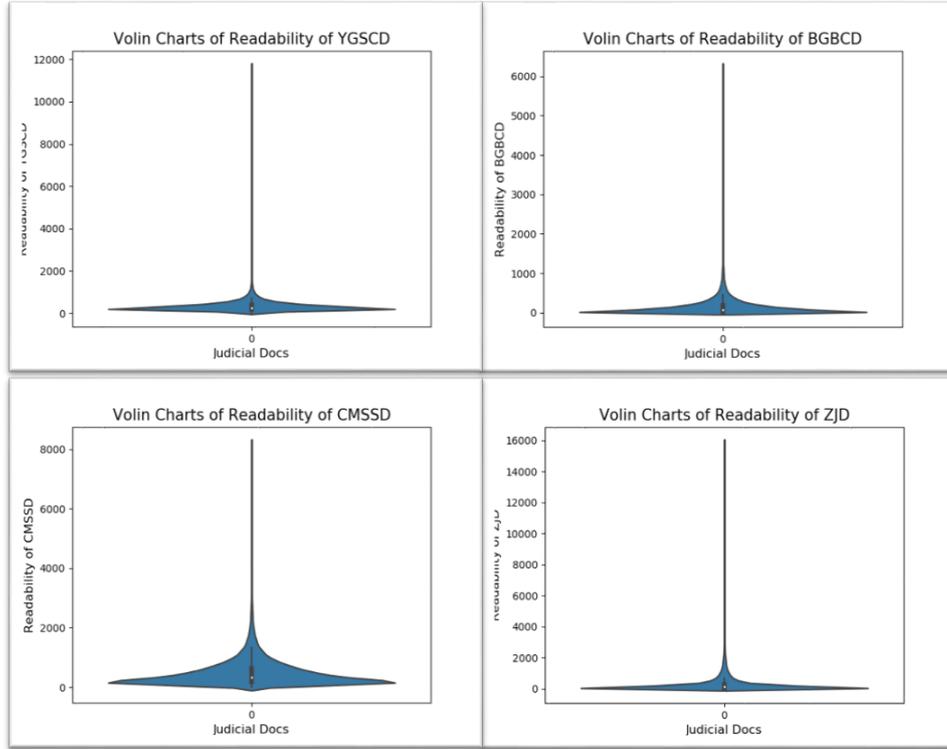


图 5.12: 文书关键段落文本长度统计 violin 图

由于要考虑系统度量指标的定制化需求，以及后续指标建设的可扩展性，所以所有度量方法均独立实现，虽然公用一个解析函数，但是后续度量计算过程互不关联。表5.2中展示了随机抽取 1000 篇文书作为输入，采用所有客观评价指标进行度量的实验结果。

5.3.2 主观度量指标的实现

1. 法条合理性

计算待测文章和库中所有文章的相似度复杂度过高，设文书库中有 m 篇文章，当待测文书有 n 篇时，计算的复杂度为 $O(m*n)$ ，这会极大的影响评估系统的即时性，所以需要先对库中案件进行划分。案由是对诉讼案件所涉及的法律关系进行概括后形成的分类名称，相同案由下的案件的诉讼争议包含的法律内容相似，所依据的法律条文自然也相似，所以我们按照最新修订版本的《民事案件案由规定》（以下简称《案由》）对库中案件进行了划分。根据划分粒度粗细的不同，《案由》共分为十大部分，43 个种类，最多四个层级，为了保留其原始

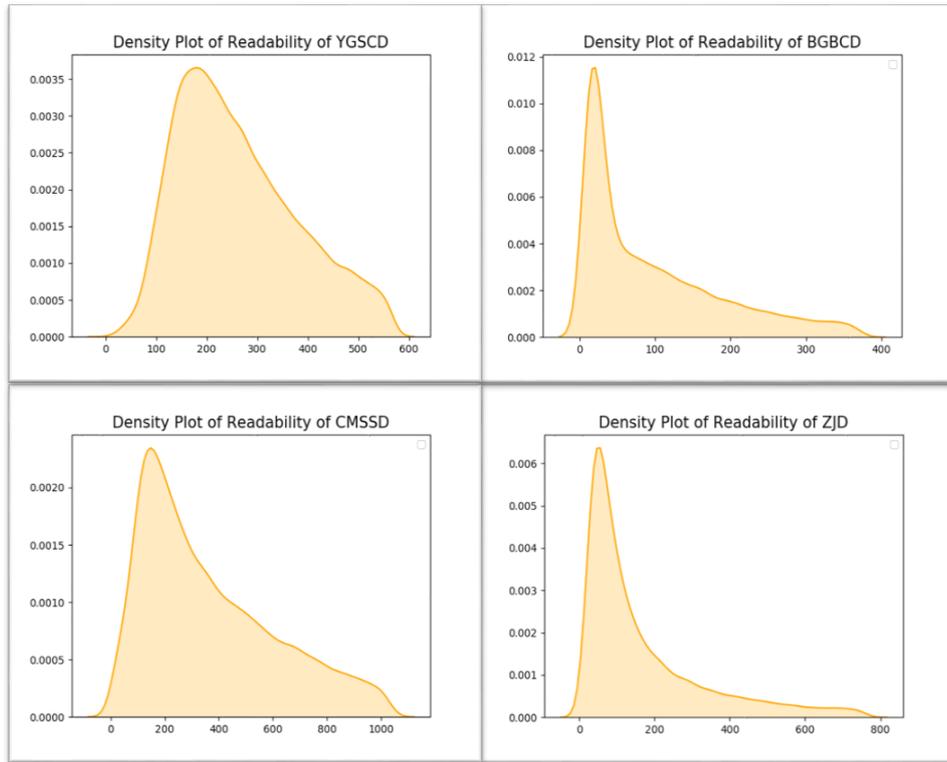


图 5.13: 文书关键段落文本长度统计密度图

的层次特征，我们在系统中构建了案由的树型结构。

根据案由下案件数量的统计我们发现，有些罕见案件的数量较少，计算相似度时选择性就比其他的案件少，甚至不能满足阈值的约束，导致推荐的相似案件其实毫不相关，所以除了按照类别划分，我们选择了 LDA 算法对其进行隐主题的划分。

无论对于案由或者 LDA 获得的文书分组，都需要在组内计算文本相似度，找到最相关的 10 例案件后汇总得到法条字典，其中包含所有的法条名称及出现次数，最后计算待测文书和相似文书中的法条字典重合程度，作为文书的法条合理性评分，并将法条字典中出现的高频法条推荐给检测者作为参考。

法条合理性度量的模型训练分为两部分，第一种是根据已有的案由文书构建案由字典，然后根据构建好的案由字典对文书进行分组，由于案由存在层级结构，对应的文书分组也存在层次关系，为了方便模型使用，在分组后按照案由规范字典对文书分组建立层级索引。第二种是根据 LDA 主题模型完成文书隐主题的训练，根据模型的评估指标，最终选取 78 作为隐主题的个数，得到文书分组和每个隐主题的主题词序列，为了加快相似文本计算的速度，本系统使用 `similarities.MatrixSimilarity.save` 将索引持久化存储在服务器中，调用索引时使用

表 5.2: 客观评价指标度量结果表

数据质量指标	最大值	最小值	平均值	标准差
参诉人信息细致性	5	3	4.91660773	0.32281001
事实部分细致性	5	5	5	0
理由部分细致性	5	0	4.45544554	1.5576369
案件信息延迟性	5	0	3.69306931	4.25602095
裁判文书延迟性	5	0	3.69306931	4.25602095
案由信息类别规范性	5	2	4.82178218	0.7091487
裁判依据引用规范性	5	5	5	0
判决内容说明完整性	5	0	0.89108911	1.91348001
诉讼费承担说明完整性	5	0	3.17821782	2.13595361
数字使用一致性	10	5	9.75247525	1.08459912
标点符号使用一致性	10	0	5.57425743	2.81912329
事实部分描述简明性	5	1	3.8019802	1.23496745
争议焦点条理性	5	0	2.52475248	2.49987746
构成事项准确性	10	6	8.1980198	1.14338218
审理经过准确性	9	2	6.45544554	1.94236129
参诉人信息准确性	3.5	0	1.54455446	1.22443968

similarities.MatrixSimilarity.load 加载索引文件即可使用。

所以法条合理性计算模块的流程为：对于输入的待测文书，系统首先要检测其是否包含案由信息，如果不包含，则直接调用 LDA 模型查找分组，如果包含案由信息，还要判断案由信息是否符合规范，也就是是否包含在构架好的案由字典中，如果不包含同样直接调取 lda 模型，包含即可按照之前构建的索引匹配案由分组文书，不同分组下包含的文书个数不同，为避免文书数量过少导致的相似度计算不准确，需要对文书数量进行审核，如果不满足阈值要求，首先应判断当前案由是否为四级案由，即详细案由，如果是则向上级递归，将上一层下所有四级节点的案件都加入到案件分组中重新衡量文书数量，如果不是四级案由，那么放入 lda 模型中进行计算，找到与之相似的文书分组。

数据预处理: 实验使用的数据集数量为 50000。对于 xml 文件，我们抽取其中的案件基本情况标签作为训练文本，裁判分析过程中的法律发条引用集合作为文本对应的发条标签。在对文本使用 TFIDF 算法转化后，过滤掉 TF-IDF 值低于 0.025 的低价值词汇。考虑到裁判文书具有的一些个性化的因素，例如案件中的姓名、时间、地址，会对案件的分类和相似度计算造成影响，而这些因素在判决过程中没有提供任何价值信息。而对于案件涉及金额的词语，应当在案件相似度计算中重点关注，我们也对其进行了聚类。具体的处理步骤如下：

1. 使用正则匹配文中涉及原告被告以及诉讼人姓名处并用空字符串替换。
2. 使用正则匹配分词结果中出现大写或小写年月日等时间点，使用空字符

串替换

3. 使用正则匹配分词结果中出现“省、市、乡、镇、州、村、区”地址信息，使用空字符串替换。

4. 使用正则匹配文中出现大写或小写形式的金额，将金额统一转换为整数格式后按区间范围进行分类。

我们将困惑度作为 LDA 实验结果好坏的评估指标，以此确定 LDA 的隐主题个数。困惑度是用来衡量语言模型对样本预测能力的好坏。当已知文本 $(w_1, w_2, w_3, \dots, w_n)$ 为语料库中文本，那么根据语言模型计算得到该文本的概率越高，则该模型对语料库的拟合程度越强。设未知测试集为 D_{test} ，困惑度的计算公式如下所示， M 为测试集中裁判文书的个数， N_d 表示第 d 篇文书中的单词个数， W_d 表示第 d 篇文书中的单词。计算得到的困惑度值越小，则模型表现越好

$$perplexity(D_{test}) = \exp\left(\frac{-\sum_{d=1}^M \log(p(w_d))}{\sum_{d=1}^M N_d}\right)$$

构建 LDA 的核心代码如图 5.14 所示：

```
#训练LDA模型
tfidf = models.TfidfModel(dict_corpora)
corpus_tfidf = tfidf[dict_corpora]
tfidf.save("./lda_model/my_model.tfidf")
np.random.seed(SOME_FIXED_SEED)
lda = models.LdaModel(corpus_tfidf, num_topics=78, id2word=dict_1, iterations=1000)
# LDA模型存储
lda.save("./lda_model/mylda_v2")
# 构建索引
index = similarities.MatrixSimilarity(lda[search_corpus])
index.save(file)
```

图 5.14: LDA 模型训练关键代码图

2. 语言风格鲜明性

实验数据准备：语言风格鲜明性的实验数据主要分为三方面，第一部分为裁判文书中的文本数据，第二部分为社会性文本类的新闻数据，第三部分为基于文本风格迁移技术生成的数据集。对于裁判文书数据，我们按照 xml 文书结构，分别抽取案件基本情况中的原告诉称段、被告辩称段及查明事实段，以及裁判分析过程中的文本，标签记为“wenshu”。样例如图所示

新闻类数据的获取比较复杂，为了保证分类模型训练的效果，训练数据需要包含司法类相关信息，避免信息内容不一致对分类模型造成的错误影响。经调查，已有新闻标签分类文本均不包含法律司法相关文本，所以采用爬虫技术

```
<YGSCD value="原告谷圣庄诉称，2014年1月10日，被告杨杰、刘成友向原告借款50000元，被告拖欠不付，请求依法判令被告偿还借款50000元及违约金，并承担本案诉讼费用。" nameCN="原告诉称段"/>
<BGBCD value="被告杨杰、刘成友均未作答辩。" nameCN="被告辩称段"/>
<CMSSD value="经审理查明，2014年1月10日，被告杨杰、刘成友向原告谷圣庄借款50000元，约定2014年2月14日还款，逾期不还，每天支付借款金额的10%违约金。经原告催要未果，故提起诉讼。以上所确认的事实，有原告的陈述笔录、原告提交的借据在案为凭，已经本院开庭审查，真实、合法，本院予以采信。" nameCN="查明事实段"/>
```

图 5.15: 文书文本数据样例图

进行爬取。爬虫的目标网站为民主与法制网，网址为 www.mzyfz.com，共获取新闻数据 5000 条，根据获取新闻包含的文本内容长度绘制的小提琴图如下，原始数据如第一张图片所示，筛除了个别较长文本后得到初始新闻数据集，计算得到数据长度的百分之九十五置信区间范围，依据置信区间对文本进行进一步筛选，得到符合条件的文本共计 4356 篇，标签记为“news”，样例如图5.16所示。

标题,内容
广东法院发布一批妨害新冠肺炎疫情防控犯罪典型案例,新华社广州2月13日电 (记者周颖) 广东省高级人民法院13日发布一批妨害 最高人民检察院依法对胡怀邦决定逮捕,新华社北京2月13日电 最高检13日消息,国家开发银行原党委书记、董事长胡怀邦涉嫌受贿 浙江首例疫情期间非法狩猎野生动物案被提起公诉,疫情防控期间,还敢捕猎野猪并宰杀。今天下午,浙江省绍兴市上虞区检察院以涉嫌 银保监会:各人身保险公司不得利用疫情事件营销,新华社北京2月13日电 (记者张千千) 中国银保监会人身险部日前发布通知,要求 山东932个法庭能够“网上开庭”,新华社济南2月12日电 (记者潘林青) 记者日前从山东省高级人民法院获悉,山东具备网上开庭功 云南省检察院对云南省司法厅原副厅长赵立功作出逮捕决定,新华社昆明2月12日电 (记者王研) 记者12日从云南省人民检察院获悉, 公安部追授牺牲在抗疫一线的郑勇二级英雄称号,2月11日,国务委员、公安部部长赵克志签署命令,追授湖北省南漳县公安局交警大 湖南公安破获贩卖伪劣口罩案 查扣“三无”口罩120余万只,新华社长沙2月11日电 (记者袁汝婷) 湖南常德鼎城区公安局10日对外 北京警方破获6起制售假冒口罩案件,新华社北京2月10日电 (记者鲁畅) 新冠肺炎疫情发生以来,北京市公安局全警动员,依法严厉 关于做好个人信息保护利用大数据支撑联防联控工作的通知,各省、自治区、直辖市网络安全和信息化委员会,中央和国家机关有关部

图 5.16: 司法类新闻文本数据样例图

第三部分数据实在前两个数据集的基础上，借用非平行式的文本风格迁移技术，抽取出新闻数据中的语言风格后，对文书数据集进行风格迁移得到新闻风格的文书数据集，标签页设置为“news”。最后在两个数据集的基础上，训练文本分类器，分类模型结构如图5.17所示：

模型主要由四层组成，卷积层是提取文本不同的 n-gram 特征。我们将内核数设置为 256，每个内核大小设置为 5。最大池层取卷积后得到的多个一维向量的最大值，然后将它们拼接在一起作为该层的输出。最后，全连接层将另一层和最大池层结合起来作为输出，以提高网络的学习能力。我们使用的模型包含两个完整的全连接层，每层由 128 个神经元组成。在迭代过程中，我们使用交叉熵作为损失函数预测惩罚，并用 AdamOptimizer 进行优化。

实验结果：训练过程的 loss 值变化如图5.18所示，训练模型在测试集上的分类准确率为百分之 95.46。

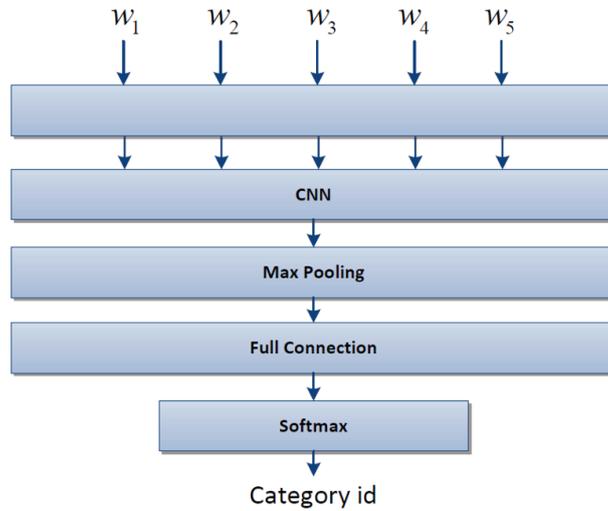


图 5.17: 基于 CNN 的文本分类模型结构图

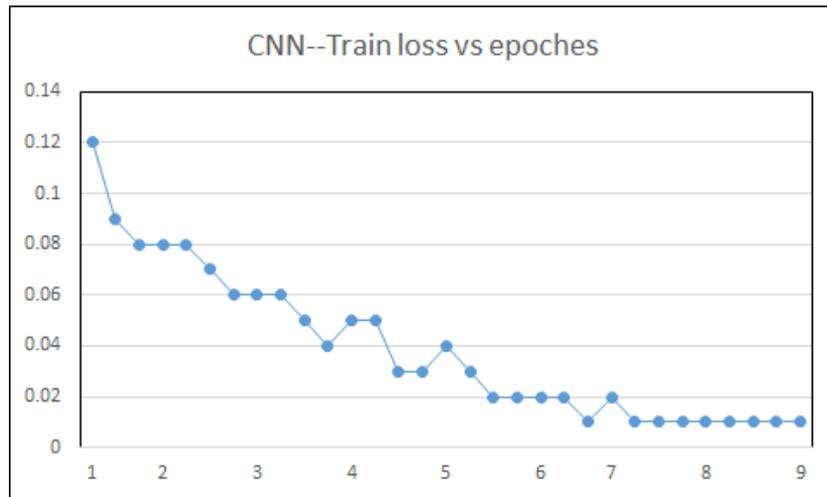


图 5.18: CNN 训练过程 loss 变化图

3. 语言情感鲜明性

1. 基于极性词的情感分析

极性词典指的是单词包含的情感色彩情况，例如“欢喜，开心”一类的词语含有积极的情感，而“厌恶，嫉妒”则具有消极的色彩，我们大脑中会保存类似的语料库，当接收到一个新的句子时，就会依据其中包含的单词对句子的情感做一个初步的分类，例如“我今天很开心”中包含“开心”，我们会认为他是具有积极色彩的语句。极性词典就是模拟人脑中的语料库，对应的基于极性词典的情感分析的处理流程如下所示：

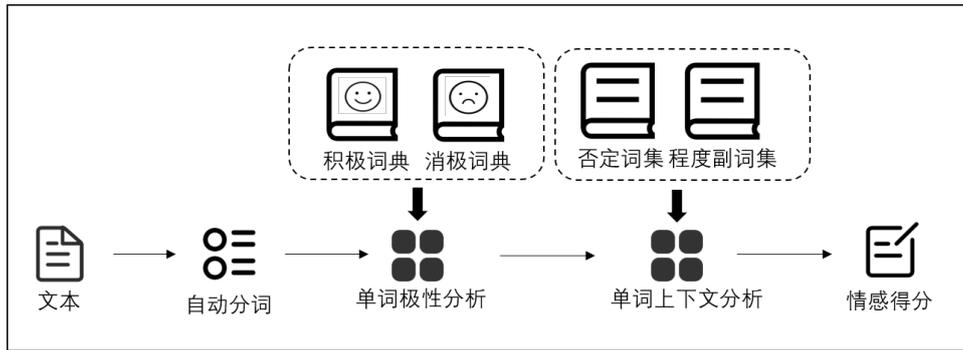


图 5.19: 基于情感极性词典的情感分析流程图

首先我们会对输入的文本进行自动分词处理，这里我们默认输入的语句已经做好了句子切分。得到单词列表后，我们需要借助标注好的词典对单词进行分类，区别积极、消极、非情感词语，以及带有情感色彩单词的情感倾向程度，对于包含情感色彩的单词，还需要判断上下文，当前文中出现程度副词时，例如“非常、稍微”，那么其后的单词的情感成都应该增加，当前文出现否定词时，例如“不，尚未”等，那么该词语就会将其后的情感词转换为相反的情感，最后我们将每个情感词的得分进行加和，得到最终完整句子的情感得分。

数据准备：根据设计部分的描述，本部分涉及的数据集包括极性词典，程度副词词典及否定词词典。极性词典使用的是来自 BosonNLP 提供的情感字典，该数据集由新闻、论坛等上百万篇情感标注数据中自动化构建得到，其中包含数据 111764 条，数据格式为单词及对应极性得分，正数为积极情感，负数为消极情感，数值的绝对值越大说明情感越强烈。为了和本文中的研究相融合，我们对其进行了过滤处理。由于司法中一些经常出现的单词在极性词典中含有较强的情感色彩，例如“盗窃罪、精神损失费”，这会对司法文本情感客观性的计算产生误导，所以我们统计了 50000 篇文书中出现的高频词汇，降低其在极性词典中的情感倾向程度，得到本文中使用的感情极性词典。除此之外，本文还使用了知网发布的情感分析用词语集和网络开源的否定词表。情感分析数据集中包含了 219 个中文程度级别词语，并对词语的程度进行了划分，详情如表 5.3 所示。否定词集中则包括了 71 个常见的否定词。

2. 基于 BERT 的情感分类算法

由于不同单词在不同的上下文中会具有不同的感情色彩，并且单句的情感也不是通过单词情感程度的线性相加结果决定的，所以需要使用深度学习的模型对情感进行评估。本文使用带标签的评论文本作为训练数据，使用基于 Bert 的分类算法训练分类模型，对于案件基本情况文本中的每一句，使用情感分类

表 5.3: 知网中文程度级别词语分类说明表

程度等级	单词数量	样例
极其/最	69	百分之百、非常、极度、最...
很	42	分外、多加、不胜...
较	37	更、还、较比...
稍	29	些许、略为、多多少少...
欠	12	轻度、没怎么、微...
超	30	出头、过于、过分

模型进行情感分析,以正负情感偏差的得分作为单句的客观性得分。BERT 模型的原理已经在相关技术中简要介绍过,在此不再详细叙述。

数据准备:本文使用开源的酒店及游戏的评论作为输入数据,其中消极标签的数据供 7379 条,积极标签的数据共 9178 条,由于司法文书中大多数为标准文本,所以对原始数据集中含有特殊符号的数据进行了过滤处理,并借用前文的极性词典对数据进行了筛选,最后对得到的数据集按照 7:2:1 的比例进行划分,得到实验使用的训练集、验证集和测试集。

实验流程:本文使用的是 BERT 开源的简体中文模型,模型一共 12 层,768 个隐藏层,110M 参数,训练时使用的参数为 $train_batch_size = 32, learning_rate = 5e - 5, num_train_epochs = 2.0$ 。本文根据数据准备阶段的数据格式,继承模型提供的 DataProcessor 类,实现 myProcessor 类完成对数据的预处理。实际使用过程中,我们使用 finetune 后的模型对文本进行分类,取分类的置信度结果作为该文本情感分类结果的评估依据。

4. 文本重构性检测

为了检测文本之间的抄袭概率,我们使用编辑距离和序列的文本相似度算法对两个待测段中的每一句进行度量。编辑距离又称 LevenShtein 距离,指的是两个字符串之间,由一个变换成另一个需要的最少编辑次数,编辑包括增删改操作,编辑距离越小则两个字符串的相似度越高,即越有可能为抄袭。序列相似性指的是忽略主体词义及语义,单纯侧重文本之间的相似度,该算法计算返回两个字符串共现序列的字符数占字符总数的比例。实际操作时取两种算法评分的均值作为每一个语句对的抄袭检测得分。设文书原告/被告段为 $Text_A$, 包含语句个数为 M , 文书查明事实段为 $Text_B$, 包含语句个数为 N , 那么对于 $Text_B$ 中的每一句话 $Text_{B_n}$ 和 $Text_A$ 中的每一句进行检测,得到最相似的语句序列:

$$\{(SIMText_{A_i}, sim_1), (SIMText_{A_j}, sim_2), \dots, (SIMText_{A_k}, sim_N)\} (i, j, k \in (0, M))$$

其中 sim_N 为抄袭概率, $SIMText_{A_k}$ 为检测指标最高的语句,最终统计抄袭

概率超过阈值的语句个数算入总分，并向用户展示对应的语句，增强检测结果的可解释性和可信度。

为了检测每个度量指标的可用性，实验使用单个指标对多篇裁判文书质量进行检测，测试输入标准格式的 xml 文件，是否能够输出数值格式的计算结果，测试使用文书 1000 篇，各个指标的输出结果情况如表格5.4所示。

表 5.4: 主观评价指标度量结果表

数据质量指标	最大值	最小值	平均值	标准差
语言风格鲜明性	0	25	15.64356436	5.23759101
情感倾向性	0	25	10.0990099	9.20019904
语言重构性	0	25	12.87128713	9.44883393
引用法条合理性	0	25	20.59405941	5.59734417

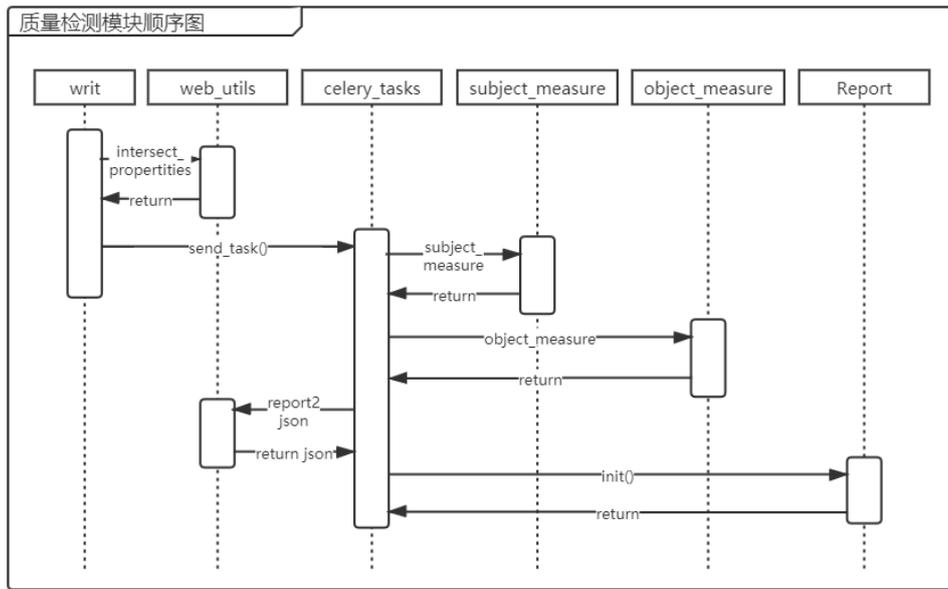


图 5.20: 文书质量检测模块顺序图

文书质量检测模块的顺序图如图5.20所示，wirt 中的路由函数 *doc_measure* 负责接受用户发起的检测请求，请求中包含文书 id 及指标配置，首先要对指标进行预处理，通过调用 *web_utils* 中的 *intersect_properties* 得到标准的指标配置字典，返回给路由函数，路由函数会调用 *celery_tasks* 中的 *send_task()* 执行异步任务，将任务存储到消息队列中等待 worker 执行，并向前端返回任务已接受正在处理的状态信息，*doc_measure* 在执行时会先后调用 *subject_measure* 和 *object_measure* 中的度量函数，完成客观和主观质量的度量，并将结果返回给

`doc_measure`。`doc_measure` 首先使用 `web_utils` 中的 `report2json` 函数将度量结果转换为 json 格式存储到文件系统中，然后根据报告信息创建 `Report` 对象，并使用 `db.session` 将对象持久化到数据库中，最后修改文书状态为检测完毕。用户在下一次查询时即可获取文书的最新状态。

单篇文书质量检测的报告页面如图5.21所示，其中主要分为文书基本信息，客观信息度量结果和主观质量度量结果三部分。基本信息包括文书标题、报告生成时间、评估得分以及评估使用的度量指标，方便用户对当前查看的文书有一个整体的了解。客观质量度量结果部分首先展示了文书的篇章完整性，由于篇章结构比较重要所以单独显示，接下来是度量结果的详细展示，按照篇章结构对文书进行划分后，详细说明文书每个片段中涉及到的具体错误以及修改意见，增强度量结果的可解释性，对于不能在文本中解释的度量指标，例如文书的延迟性，会在报告中以进度条的形式单独列出，供用户查看，最后会将详细的指标按照类别进行统计，统计结果以雷达图的形式呈现，方便用户直观了解文书在每个度量维度的表现；主观质量度量结果将四个指标都用可解释的方式呈现出来。抄袭检测如果涉及抄袭语句会直接指出，语言风格鲜明性和文本客观性则会以文本颜色的深浅表示，对于有问题的句子会以其他的颜色标明，并且颜色越鲜艳则质量越差；引用法条合理性会将计算的中间结果中相似案例的高频法条推荐给用户。用户可以在报告开头选择下载 pdf 留存质量度量结果。



图 5.21: 系统单篇文书质量报告页面

5.4 统计分析模块实现

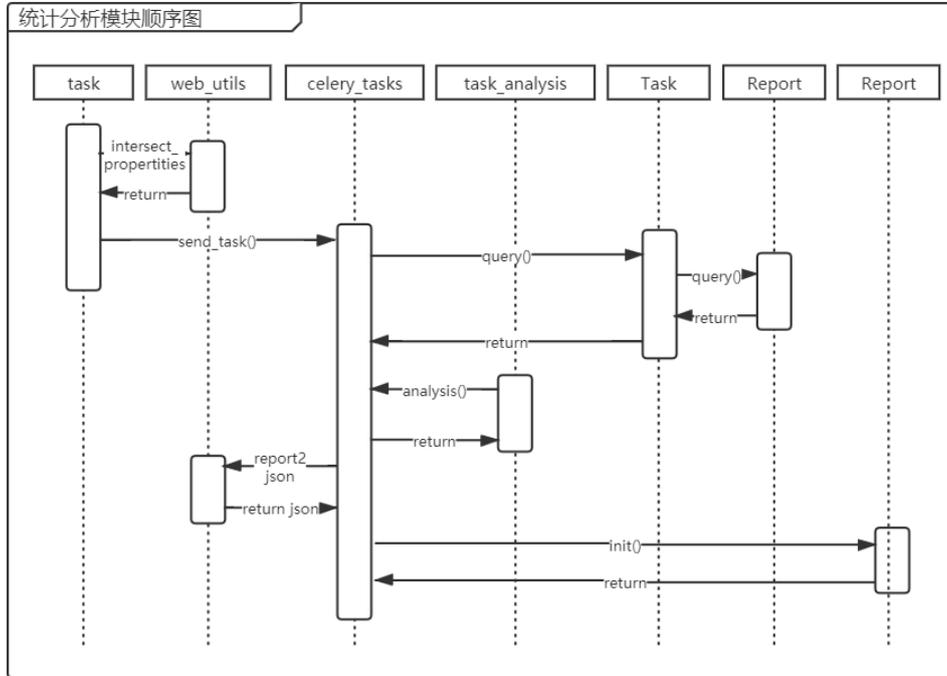


图 5.22: 统计分析模块顺序图

如图5.22为文书质量统计分析模块的顺序图。*task* 中的 *task_measure* 负责处理前端发起的统计分析请求, 请求中包含任务 id 和指标配置。和文书质检模块类似, 首先调用 *web_utils* 中的 *intersect_properties* 函数获得标准字典后返回, 然后 *task* 会调用 *celery_tasks* 中的 *send_task()* 调用异步任务, 将任务加入消息队列中并将接受请求信息返回给前端, *task_measure* 执行时首先会根据任务 id 查找对应的 *Task* 对象, 通过 *Task* 对象获取任务中包含的文书 id 列表, 根据文书 id 查找对应的 *Report* 对象, 得到任务中所有文书的检测结果文件地址及省份, 年份等结构化信息, 返回给 *task_measure* 函数, 接着函数会获取单篇质检文书的结果进行统计分析, 得到统计分析结果后交给 *web_utils*, 工具类中的 *task_report2json* 会将结果转换为定义好的 json 格式返回给 *celery_task*, 最后创建 *AnalysisReport* 对象持久化存储报告内容, 并将任务状态修改为已检测完成。

统计报告的可视化界面如图5.23所示, 主要分为任务基本信息, 数据统计分析以及度量结果分析三部分。第一部分包括任务的名称、创建时间、包含任务数量、平均得分以及度量使用的指标组合, 方便用户对任务的基本情况和度量整体结果有一个了解。第二部分展现了任务中包含文书的时间和地域信息的统



图 5.23: 系统批量检测统计分析报告页面

计结果，最后一部分呈现了度量总结结果的分析，包括客观质量维度图，和每个详细指标的均值与标准均值之间的差异分析。用户可以在报告开始选择下载 pdf 版本的可视化报告，也可下载 json 格式的数据进行后续处理，同时，用户也可以查看任务中包含的所有文书的单篇检测报告详情。

5.5 本章小结

本章负责系统实现情况的详细介绍，按照第四章评估系统设计中的模块划分，系统实现也对应分为数据交互模块、数据管理模块、文书质量检测模块以及统计分析模块四部分，使用顺序图对各个功能模块中的调用关系进行了说明，同时展示了涉及到的系统页面。其中在文书质量检测模块中详细阐述了各个度量指标的实现方法及度量结果。

第六章 民事裁判文书评估系统测试与分析

6.1 测试目的

系统测试为检验本系统是否满足第三章提出的需求及设计部分要求，本小节设计测试用例对文书质量检测系统进行测试。测试部分主要分为系统功能性测试和性能测试。

6.2 测试环境

系统测试使用的环境如表格6.1所示。用户计算机为客户端，负责与平台进行交互，前端展示服务器负责平台页面的展示和 web 层面交互。应用服务器负责响应前端发起的请求以及数据持久化存储，计算服务器负责文书质量度量及统计分析的计算。

表 6.1: 系统测试环境说明表

设备与软件	环境及配置说明
用户计算机	Win10, Chrome77.0(64 位)
前端展示服务器	Ubuntu 16.04 LTS, Node.js v10.12.0, Nginx 1.17.9, CPU 2 核, 内存 4G, 硬盘 500G
应用服务器	Ubuntu 16.04 LTS, Python3.5 及以上, Flask1.1.1, SQL57, Redis3.0, CPU 2 核, 内存 4G, 硬盘 500G
计算服务器	Ubuntu 16.04 LTS, Python3.5 及以上, CPU 2 核, 内存 4G, 硬盘 500G
部署软件	Jenkins 2.222 LTS

6.3 系统功能测试

按照第三章中阐述的需求分析及设计内容，本小节从用户的角度出发，模拟用户上传文书，数据管理，质量检测到最终得到文书质量检测结果的操作，根据需求分析中的功能说明设计测试用例，检测完整系统的可用性和功能点覆盖程度是否符合要求，详细测试用例描述表及测试内容说明如下：

表6.2为用户数据上传测试用例，测试用户是否可以正常上传单篇文书以及文书压缩包，后端是否能够实现自动过滤文件类型并完成存储，以及存储信息是否正确。

表 6.2: 用户数据上传测试用例描述表

测试 ID	TC1
测试名称	用户上传文书功能测试
测试功能	系统是否能正确响应上传文书请求，并将上传文书存储至服务器文件系统，信息更新至数据库
测试步骤	<ol style="list-style-type: none"> 1. 用户点击上传文书按钮 2. 在弹出的信息框中点击“选择文书”按钮 3. 在本地选择单篇文书或文书压缩包 4. 确认上传列表无误后点击上传按钮
预期效果	后端接收到用户的 id 和上传请求后，会自动对上传的文件进行解压和过滤等操作，只保留 xml 文件存储到文件系统，并将文件信息存储至数据库，然后将保存成功的文件信息返回给前端，用户可以在我的列表中查看新增的文件，文件对应的状态应为未检测

表6.3为用户数据下载测试用例描述，用户数据下载包括单篇文书的结果下载和批量文书的数据下载，该测试用例检测两种情况下用户是否能够正确获取指定格式的检测结果，

表 6.3: 用户数据下载测试用例描述表

测试 ID	TC2
测试名称	用户结果下载功能测试
测试功能	系统是否能正确响应下载结果请求，以及下载内容是否正确
测试步骤	<ol style="list-style-type: none"> 1. 用户在文书界面点击下载 pdf 报告按钮或者下载数据按钮 2. 用户选择文件下载地址 3. 文件下载到用户指定位置
预期效果	如果用户选择了 pdf 格式报告下载，得到的结果应为报告的展开格式，如果下载数据为 json 格式文件，单篇文书的数据文件中除了包含可视化报告中展示的内容，还包含度量用到的每个指标的具体得分，批量文书的数据文件中包含每篇文书的结果和最终的汇总统计数据

表6.4描述了用户配置管理测试用例，测试用户是否可查看和修改当前指标配置，并在一键检测时使用默认配置完成检测。

表 6.4: 用户配置管理测试用例描述表

测试 ID	TC3
测试名称	用户配置管理功能测试
测试功能	用户能够查看和修改当前配置，并可在发起检测任务时使用默认配置
测试步骤	<ol style="list-style-type: none"> 1. 点击我的配置按钮进入配置界面 2. 用户查看当前配置，详细显示每个度量指标以及度量含义 3. 用户更改配置并点击保存按钮 4. 文书检测或建立检测任务前，用户可点击一键检测按钮使用默认配置完成检测
预期效果	<ol style="list-style-type: none"> 1. 当前端发起用户查看配置请求时，从数据库中调取当前用户的配置信息返回前端，配置信息包含指标名称、解释及当前状态 2. 当用户发起修改请求后，修改数据库中存储的配置信息，并将修改结果返回给前端，页面提示用户修改成功 3. 用户选择默认配置时自动读取数据库中保存配置完成文书检测

表6.5为文书管理测试用例描述表，目的在于检测系统是否实现文书管理功能，功能包括文书列表的查看，以及根据名称和日期对文书列表进行检索。

表 6.5: 用户文书管理测试用例描述表

测试 ID	TC4
测试名称	用户文书管理功能测试
测试功能	用户能否对已上传文书列表进行管理
测试步骤	<ol style="list-style-type: none"> 1. 用户点击我的文书按钮，查看当前文书列表 2. 在输入框中输入文书名称，点击搜索按钮查看过滤后的文书列表 3. 在输入框中输入起始时间和终止时间，点击搜索按钮，查看过滤后的文书列表
预期效果	<ol style="list-style-type: none"> 1. 用户可查看所有已上传文书的相关信息，信息项包括文书文件名称，文书大小以及当前状态，状态分为待检测，已完成及检测中 2. 系统可根据用户输入的文件名称进行模糊匹配，返回所有匹配的文书列表 3. 用户可以输入单个时间节点或两个时间节点作为筛选条件，系统会返回所有匹配的文书列表

表6.6为用户任务管理用例描述表，系统为批量文书检测建立任务管理机制，本用例测试系统是否满足 UC5 中描述的任务管理功能要求。

表6.7为文书质量检测功能的测试用例描述，由于文书预处理对用户为隐藏功能，同时也是文书质量检测的前提条件，所以只对文书质量检测进行测试，测试内容包括单篇文书的客观和主观质量度量。

表 6.6: 用户任务管理测试用例描述表

测试 ID	TC5
测试名称	用户任务管理功能测试
测试功能	系统提供批量文书创建任务功能，并支持用户查看任务列表及当前状态
测试步骤	<ol style="list-style-type: none"> 1. 用户在我的文书列表批量点选文书，然后单击创建任务按钮 2. 用户在弹出的信息框中选择任务的基本信息，包括任务名称及检测指标配置，像后端发起建立任务请求 3. 用户在我的任务界面查看已有检测任务的基本信息，包括任务名称，包含文书个数、建立时间以及当前检测状态 4. 用户点击刷新按钮实时获得任务的最新状态
预期效果	<ol style="list-style-type: none"> 1. 用户发起创建任务请求后，后端自动对任务中包含的文书进行质量检测 and 统计分析，并在任务完成后更改任务状态 2. 前端发起查看任务列表请求时返回当前用户创建的任务信息 3. 用户点击刷新按钮时获取当前数据库中存储任务信息返回给前端

表 6.7: 文书质量检测测试用例描述表

测试 ID	TC6
测试名称	文书质量检测测试
测试功能	系统对输入的单篇文书可以完成质量检测并输出前端要求的结果数据
测试步骤	<ol style="list-style-type: none"> 1. 用户在我的文书列表中选择要检测的文书单击开始检测 2. 用户选择检测需要的度量指标，或者使用默认配置完成检测，前端将配置和文书 id 发送给后端发起检测请求 3. 用户点击查看报告按钮显示可视化报告
预期效果	<ol style="list-style-type: none"> 1. 后端在收到请求后自动完成文书的预处理工作，包括关键信息抽取和文本预处理，然后交给度量模块进行度量 2. 获取用户输入配置或数据库中存储默认配置 3. 检测完成修改文书状态，用户刷新时可观察到文书状态由未检测变为已完成检测 4. 将检测得到的数值结算结果及文本结果生成指定 json 格式传送给前端进行界面展示，并在后端存储

表6.8为质检结果统计分析功能测试用例描述，目的为检测系统能否实现 UC8 表中功能。

表 6.8: 质检结果统计分析功能测试用例描述表

测试 ID	TC7
测试名称	质检结果统计分析功能测试
测试功能	系统对于任务级文书检测结果能否进行汇总分析, 得到指定格式的分析数据
测试步骤	1. 用户在我的列表中选择文书后建立任务, 选择检测指标或使用默认指标发起检测请求 2. 用户点击检测完成的任务查看统计分析结果
预期效果	1. 后端在收到请求后首先进行单篇文书质量检测, 在此基础上完成统计汇总, 并将任务状态修改为已完成 2. 将统计数据存储至后端在用户发起请求时传送给前端, 前端渲染后呈现给用户

6.4 系统性能测试

为保障民事裁判文书质量评估平台设计与开发的实用性与可靠性, 本小节对系统进行了性能测试, 测试内容包括时间特性和资源利用性两部分, 详细测试项如表6.9所示:

表 6.9: 系统性能效率测试用例描述表

测试项目	测试要求	
时间特性	响应时间	对前端发起的响应请求时间不超过 1s
	输入时间	用户将一百篇裁判文书上传至系统并存储的时间不超过 30s
	运行时间	各项指标在 100 篇司法文书上进行度量的平均时间不超过 90s
	输出时间	系统将 100 篇文书的度量结果写入文件花费的时间不超过 6s
资源利用性	案由类型利用率	测试用例中司法文书案由类型覆盖率达到 100%
	案件数目利用率	民事审判类文书的覆盖率达到 100%
	文书关键字段类型利用率	司法文书关键字段类型利用度达到 70%

6.5 测试结果

表6.10中展示了系统功能测试的结果, 测试人员按照表中内容操作, 观察后端对请求的相应情况是否满足预期效果, 并对实验结果进行了记录。系统的度量耗时实验结果如图6.1所示, 客观耗时的平均时间为 0.02s, 主观指标由于使用了深度学习或机器学习的模型, 相对耗时会比较长, 串行运行所有指标的平均

表 6.10: 系统测试用例执行结果表

测试用例 ID	对应用例描述 ID	测试结果
TC1	UC1	通过
TC2	UC2	通过
TC3	UC3	通过
TC4	UC4	通过
TC5	UC5	通过
TC6	UC7	通过
TC7	UC8	通过
TC8	—	通过

耗时为 3.58s，但由于指标之间计算相互独立，所以在实际使用过程中可以开启多线程并行运行，所以实际耗时不会超过 1s，同时文书与文书之间的度量可以并行运行，不存在抢占资源的情况，对于异步度量任务，以上耗时完全可以满足用户需求。其他测试结果如表6.10所示，系统满足了需求分析中定义的所有功能需求，并且性能也符合实际应用需求，具有实际的开发意义。

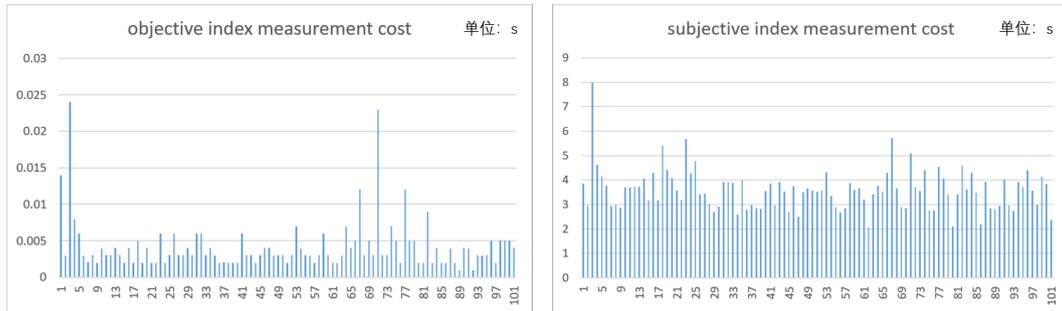


图 6.1: 系统度量耗时统计分析

并且为了保证系统质检结果的可靠性，本文采取人工验证的方法进行验证，随机抽取 15 篇文书的部分指标进行人工评分，参与人员包括司法专业人员和非司法专业人员两种类型，共收集 128 例度量结果，其中分类型指标的命中率达到百分之 84.4，数值类指标的 Spearman 系数为 0.92，Kendall 系数为 0.83，充分验证了系统度量结果可靠性，数值类指标的详细表现如表6.11所示，可以看到判决内容完整性得分较低，详细分析后发现是由于系统评估过程中对于某些告知事项要求过于严格，忽略了不同编写者的书写规范不同，而人工评审相对要求较低，导致二者分数出现了偏差，所以修改后放宽了匹配规则，采用模糊匹配的模式，但对于详细的涉及到法律名称的内容不做修改。

表 6.11: 人工验证实验结果表

	构成事项准确性	判决内容完整性	标点使用一致性	审理经过准确性
系统评分均值	8.0	4.81	5.75	6.438
人工评分均值	8.5	3.875	6	6.875
最大差值	-2	2	1	-3
差值方差	1.25	0.93	0.5	0.746
Kendall 系数	0.688	0.441	0.913	0.870
Spearman 系数	0.730	0.481	0.965	0.930

6.6 本章小结

本章主要从系统功能性需求和非功能性需求两个角度对文书质量评估平台进行了测试。功能测试的测试用例对照第四章中系统用例描述表的功能进行设计，非功能性测试包括系统的时间特性和资源利用性，分别衡量系统使用时各个阶段的延迟时间和对司法数据的覆盖程度。测试结果显示系统均通过所有测试用例，表明质量度量平台具有一定的实用性和可靠性。

第七章 总结与展望

7.1 总结

随着智慧法院建设工作的推进，越来越多的司法数据开始使用电子化的方式进行存储和流通，但数据数量的急剧增长也为传统的人工质量度量方式带来了挑战，本文以民事裁判文书数据为例构建质量评估平台，实现全流程自动化且可解释的文书质量度量。

本文的主要工作如下：本文首先调研了已有的信息及数据质量度量的方法，总结归纳了信息质量及数据质量领域常用的度量维度和度量指标，为裁判文书质量度量提供理论基础，并且对系统中使用到的算法和技术进行了调研，为系统选择适合使用的方法做准备。

然后进行了度量指标的设计。借鉴信息质量的度量方法，本文将裁判文书的质量度量划分为主观和客观两方面，客观度量中主要以裁判文书书写规范相关文件为参考，度量文书表层内容规范性；主观质量借助自然语言处理领域的研究成果，结合文书文本独有的特征选择合适的方法模拟人工度量的过程，完成文书中文本深层语义信息的度量。

接着是评估平台的系统设计，在明确系统边界后将系统划分为数据交互模块、用户数据管理模块、质量检测模块以及统计分析模块，在此基础上对每个模块中的详细功能需求进行了描述，然后从多个视图的角度对系统整体进行概要设计，接着使用类图对每个功能模块进行详细设计说明，最后对使用到的数据库结构进行了详细说明。

最后完成系统的实现与测试。实现部分分为指标计算实现和系统实现两部分。计算指标部分使用流程图及核心代码等说明指标度量的实现过程，并引用真实文书数据计算得到的结果作为验证。系统实现部分通过顺序图及系统实际页面截图展现文书评估平台的实现结果。为了保障系统实现功能的可行性和可靠性，本文设计了详细的功能性和非功能性需求对系统进行测试。

7.2 未来工作与展望

面向民事裁判文书的质量度量平台是为了实现全自动化的文书质检流程，并将检测结果可视化的展示给用户，目前实现的系统已基本满足预期设想，但由于本系统的目标是与司法公开平台进行集成，所以还有许多可以优化的地方：

文书类型未达到全覆盖。由于司法领域中不同案件的裁判文书书写格式规范不同，并且许多类型的文书书写样式规范文件不健全，所以本文中的文书质量客观度量指标只适用于民事判决书，对于其他类型的文书不适用，后续应该投入更多的人力进行司法相关信息的调研和整理，丰富客观度量中使用的规则，达到文书类型的百分之百覆盖率。

对海量数据质量检测处理的能力不足。由于本系统未使用分布式架构进行设计，无法实现海量数据的并行计算。但由于系统中度量指标设计时考虑了后期扩展及部署，实现时均相互独立，互不干扰，并且使用的 Celery 支持分布式任务调度，也为后期系统的迁移做准备，后续可以考虑使用 HDFS 替代服务器文件系统，Hive 替代 MySQL 数据库系统，并采用 Hadoop 或 Spark 分布式计算框架。

与已有公开司法业务平台的对接。由于本系统目前使用的是 xml 文件作为数据输入，与现有的文书公开平台提供的 pdf 数据格式不一致，所以后期如果对接需要先完成文书解析，或者和已有的解析软件进行对接，然后由解析软件再和裁判文书网进行对接，实现全自动的文书质量监测功能，这个也是本项目接下来的主要工作。

参考文献

- [1] 葛某智, Markus Helfert, 苏颖, 信息质量研究的现状与未来研究方向, 中国科技资源导刊 (1) (2009) 10–21.
- [2] T. Shen, T. Lei, R. Barzilay, T. Jaakkola, Style transfer from non-parallel text by cross-alignment.
- [3] 胡云腾, 论裁判文书的说理, 法律适用 (03).
- [4] 张慧清, 关俭科, 从量大法系判决释理谈我国裁判文书改革, 佛山科学技术学院学报 (社会科学版) (05) (2006) 82–85.
- [5] 韩玉刚, 裁判文书说理问题研究 (2013).
- [6] 张彩红, 裁判文书说理的评价指标体系 (2016).
- [7] 杨贝, 裁判文书说理的量化评价——以 2017 年北京市判决书论证质量调查为例, 中国应用法学 (2) (2018) 17–29.
- [8] G. S. G. She, Y. Z. Y. Zhang, Research on proofreading system of judicial documents based on n-gram model, 2011.
- [9] 李嘉伟, 面向法院裁判文书的质量检测技术研究 (2018).
- [10] 廉昊, 面向裁判文书的大数据质量评估平台的设计和实现 (2019).
- [11] R. Y. Wang, D. M. Strong, Beyond accuracy: What data quality means to data consumers, Journal of Management Information Systems 12 (4) 5–33.
- [12] D. Aebi, L. Perrochon, Towards improving data quality., in: CiSMOD, Citeseer, 1993, pp. 273–281.
- [13] Y. Wand, R. Y. Wang, Anchoring data quality dimensions in ontological foundations, Communications of the ACM 39 (11) (1996) 86–95.
- [14] R. Y. Wang, H. B. Kon, S. E. Madnick, Data quality requirements analysis and modeling, in: Proceedings of IEEE 9th International Conference on Data Engineering, IEEE, 1993, pp. 670–677.

- [15] J. E. Alexander, M. A. Tate, Web wisdom: How to evaluate and create web page quality, L. Erlbaum Associates Inc., 1999.
- [16] X. Zhu, S. Gauch, Incorporating quality metrics in centralized/distributed information retrieval on the world wide web, in: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, 2000, pp. 288–295.
- [17] S.-a. Knight, J. Burn, Developing a framework for assessing information quality on the world wide web., *Informing Science* 8.
- [18] J. J. Cao, Research on some basic problems in data quality control (2010) 1–12.
- [19] W. Zong, F. Wu, The challenge of data quality in the big data age, *Journal of Xi'an Jiaotong University (Social Sciences)* 33 (5) (2013) 38–43.
- [20] L. Cai, Y. Zhu, The challenges of data quality and data quality assessment in the big data era, *Data science journal* 14.
- [21] R. Y. Wang, M. Ziad, Y. W. Lee, *Data quality* 23 (3) (1997) 11–11.
- [22] K. Huang, Y. Lee, R. Wang, Quality information and knowledge management.
- [23] C. W. Fisher, B. R. Kingma, Criticality of data quality as exemplified in two disasters, *Information Management* 39 (2) (2001) 109–116.
- [24] Improving financial services data quality –a financial company practice, *International Journal of Lean Six Sigma* 6 (2) 98–110.
- [25] R. Jugulum, R. Jugulum, Competing with high quality data: concepts, tools and techniques for buliding a successful approach to data quality.
- [26] 马费成, 信息管理学基础, 2005.
- [27] 张博, 宋立荣, 农业科技信息共享中信息质量需求分析, *中国农学通报* 26 (10) (2010) 343–346.
- [28] B. Klein, Data quality in the practice of consumer product management: Evidence from the field., *Data Quality Journal* 4.

- [29] L. A. Gatys, A. S. Ecker, M. Bethge, Image style transfer using convolutional neural networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- [30] O. T. Ming-Yu Liu, Coupled generative adversarial networks, *Advances in Neural Information Processing Systems* (2016) 469–477.
- [31] J. Y. Zhu, T. Park, P. Isola, A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks.
- [32] J. Mueller, D. Gifford, T. Jaakkola, Sequence to better sequence: Continuous revision of combinatorial structures, in: D. Precup, Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70 of *Proceedings of Machine Learning Research*, PMLR, International Convention Centre, Sydney, Australia, 2017, pp. 2536–2544.
- [33] Q. V. Le, T. Mikolov, Distributed representations of sentences and documents.
- [34] D. Zhang, H. Xu, Z. Su, Y. Xu, Chinese comments sentiment classification based on word2vec and svmperf, *Expert Systems with Applications* 42 (4) (2015) 1857–1863.
- [35] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, arXiv preprint arXiv:1802.05365.
- [36] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding.
- [37] W.L. Taylor, “cloze procedure”: A new tool for measuring readability, *Journalism quarterly* 30 (4) (1953) 415–433.

简历与科研成果

基本情况

黎宇，女，汉族，1996年9月出生，辽宁省大连市人。研究生期间随导师参加“多元智能化诉讼服务及审判执行关键技术研究”国家重点研发计划项目，负责课题五“人民法院业务和数据标准研究和制定、司法基本服务库和案例筛选评估模型构建”中关于司法基本服务库文书质检服务的研究工作。

教育背景

2018.9 ~ 2020.6	南京大学软件学院	硕士
20014.9 ~ 2018.7	大连理工大学软件学院	本科

论文

1. Tieke He, **Yu Li**, Zhipeng Zou, Qing Wu:L2R-QA: An Open-Domain Question Answering Framework. IScIDE (2) 2019: 151-162
2. **Yu Li**, Tieke He, Ge Yan, Shu Zhang, Hui Wang:Using Case Facts to Predict Penalty with Deep Learning. ICPCSEE (2) 2019: 610-617
3. Baogui Chen, **Yu Li**, Shu Zhang, Hao Lian, Tieke He: A Deep Learning Method for Judicial Decision Support. QRS Companion 2019: 145-149
4. Hui Wang, Tieke He, Zhipeng Zou, Siyuan Shen, **Yu Li**: Using Case Facts to Predict Accusation Based on Deep Learning. QRS Companion 2019: 133-137
5. Shu Zhang, Ge Yan, **Yu Li**, Jia Liu: Evaluation of Judicial Imprisonment Term Prediction Model Based on Text Mutation. 62-65
6. Ge Yan,**Yu Li** , Siyuan Shen, Shu Zhang, Jia Liu: Law Article Prediction Based on Deep Learning. 281-284
7. Ge Yan, **Yu Li**, Shu Zhang, Zhenyu Chen: Data Augmentation for Deep Learning of Judgment Documents. 232-242

专利

1. 何铁科，黎宇，邹志鹏，顾宇，陈振宇，史洋洋，”一种使用神经网络和机器学习排序算法的问答系统实现方法”，申请号：201811298287.5，已受理。
2. 何铁科，黎宇，张舒，刘嘉，严格”一种基于案情事实的深度刑期预测方法”，申请号：201910352428.5，已受理。

致 谢

本文在此对所有在研究生学习及生活期间帮助过我的人致谢。

首先要感谢我的研究生导师何铁科老师，他在我初涉科学研究领域期间为我提供了很多指导和建议，让迷茫的我找到了研究和努力的方向，同时教会我们科研任务不但需要个人艰苦的奋斗，也需要思想上“与时俱进”，时刻关注研究领域最新的研究成果；工作上“集思广益”，定期请教团队内其他成员的最新学习成果和感悟，最后也需要健康的身心作为一切工作的基础保障。

同时，要感谢南京大学智能软件工程实验室，感谢陈振宇老师的耐心指导。实验室是研究生阶段我最有归属感的地方，为我提供了大量的学习资源和实践机会，帮助我们完成学生到职场人的身份转变，并且教会我们无论在任何岗位都要认真负责，有所担当。还要感谢实验室中结识的同学们，无论是学长学姐，同届的战友，还是学弟学妹，在日常学习和毕业论文完成阶段都给予了我很大的帮助，我从他们的身上不但收获了大量的专业知识，也学到了善良与勤奋。

感谢我在华为实习期间所有给予过我帮助的同事们，很幸运在刚刚进入职场时就遇到你们，谢谢你们在技术上对我的指导和帮助，以及在职业规划中为我提出的宝贵意见。

还要谢谢我的家人，由于疫情影响我的毕业论文大部分为居家完成，感谢我的父母为我的提供了良好的学习环境，让我能够专心完成毕业论文的所有工作，并且在答辩和视频等准备工作上为我提供了很多宝贵的意见。也要感谢我的朋友，在我论文和实验遇到困难时给予我莫大的鼓励和力所能及的帮助。

最后感谢评审和参与答辩的各位老师为我论文提出的修改意见，让我能够不断进步，谢谢各位老师的辛勤工作。

版权与原创性说明

任何收存和保管本论文的单位和个人，未经作者本人授权，不得将本论文转借他人并复印、抄录、拍照或以任何方式传播，否则，引起有碍作者著作权权益的问题，将可能承担法律责任。

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含其他个人或集体已经发表或撰写的作品成果。本文所引用的重要文献，均已在文中以明确方式标明。本声明的法律结果由本人承担。

作者签名：



日期: 2020 年 5 月 18 日

《学位论文出版授权书》

本人完全同意《中国优秀博硕士学位论文全文数据库出版章程》(以下简称“章程”),愿意将本人的学位论文提交“中国学术期刊(光盘版)电子杂志社”在《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》中全文发表。《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》可以以电子、网络及其他数字媒体形式公开出版,并同意编入《中国知识资源总库》,在《中国博硕士学位论文评价数据库》中使用和在互联网上传播,同意按“章程”规定享受相关权益。

作者签名:  黎宇
2020年5月26日

论文题名	面向民事裁判文书质量评估平台的设计和实现				
研究生学号	MF1832077	所在院系	软件学院	学位年度	2020
论文级别	<input type="checkbox"/> 学术学位硕士 <input checked="" type="checkbox"/> 专业学位硕士 <input type="checkbox"/> 学术学位博士 <input type="checkbox"/> 专业学位博士 (请在方框内画钩)				
作者 Email	liyu9691@qq.com				
导师姓名	刘嘉 副教授, 何铁科 助理研究员				

论文涉密情况:

不保密

保密, 保密期(____年____月____日至____年____月____日)