南京大学

研 究 生 毕 业 论 文
（申请工程硕士学位）

论　文　题　目　数据质量驱动的文书可解释性分析系统设计与实现

作　者　姓　名　张朱佩田

学科、专业名称　工程硕士（软件工程领域）

研　究　方　向　软件工程

指　导　教　师　陈振宇 教授，冯洋 助理研究员

2021 年 **5** 月 **19** 日

学　　号：**MF1932250**

论文答辩日期：**2021** 年 **5** 月 **20** 日

指 导 教 师：　　（签字）

The Design and Implementation of Data-Quality Driven Interpretability Analysis System for Judgement Documents

by

Peitian Zhangzhu

Supervised by

Professor Zhenyu Chen, Assistant Researcher Yang Feng

A dissertation submitted to
the graduate school of Nanjing University
in partial fulfilment of the requirements for the degree of
MASTER OF ENGINEERING
in
Software Engineering

Software Institute
Nanjing University

May 19, 2021

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：数据质量驱动的文书可解释性分析系统设计与实现

工程硕士（软件工程领域）专业 2019 级工程硕士生姓名：张朱佩田

指导教师（姓名、职称）：陈振宇 教授，冯洋 助理研究员

摘　　要

近年来，我国在智慧法院建设上有了重大进展，但还存在许多问题亟待解决，有较大的发展空间。司法数据作为智慧法院建设中不可缺少的一环，扮演着重要角色。其中，裁判文书是法院履行审判职责、展示司法正义的载体。由于全球范围内民商事案件始终保持增长趋势，且在互联网盛行时代下的裁判文书透明度不断增大，无论是数量还是质量都广受关注。在如此庞大的数量下，如何控制民事裁判文书质量以及利用好文书数据传达的信息有待研究。

本文针对裁判文书篇幅过长、裁判文书集海量数据信息繁杂冗余等问题，提出数据质量驱动的裁判文书可解释性分析技术，力求提高数据价值密度。该技术突破篇幅限制并充分发掘数据蕴含价值，融合司法规则，应用文本摘要、依存句法分析等前沿技术构建事实模型和多维度量体系。多维度量体系共涵盖了规范性、完备性、可读性、响应性、延时性、丰富性六个维度和二十四个指标。通过推动司法规则和前沿技术协同，保证裁判文书准确性及利用率，构建问题列表检测文书质量。

为降低裁判文书使用者的时间成本，且在最大化获取文书信息的同时减少为提高数据质量而进行数据处理的工作量，本文以分析服务作为技术核心，在低成本高效率的操作基础上，分别从细粒度和粗粒度层面（二者分别面向单篇文书和文书集）设计事实模型和构建基于数据/信息质量的多维度量体系。利用自然语言处理技术自动化处理分析数据，呈现数据质量驱动的文书及文书集可解释性分析报告，同时检测并定位文书问题，提供修复建议。基于本技术开发了裁判文书分析系统，包括数据交互模块、度量解析模块、质量提升模块，提供用户友好的交互界面，帮助用户轻松解析数据，感知代表性信息。

关键词： 数据质量；可解释性；文本摘要；质量提升

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 Data-Quality Driven Interpretability Analysis System for Judgement Documents

SPECIALIZATION: Software Engineering

POSTGRADUATE: Peitian Zhangzhu

MENTOR: Professor Zhenyu Chen, Assistant Researcher Yang Feng

Abstract

In recent years, our country has made significant progress in the construction of smart courts. However, there still exsit many problems that need to be solved urgently. Judicial data, as an indispensable part, plays an important role in the construction of smart courts. The judgment document is the carrier for the court to perform its judicial duties and demonstrate judicial justice. Due to the continuely increasing trend in the civil and commercial cases as well as the growing transparent of judicial justice, the quantity and quality of judgment documents have attracted more and more attention. In a large number of judgment documents, it is an important challenge about how to control the quality of documents and make good use of the information conveyed by these documents remains to be studied.

To alleviate the problem of excessive length and redundant information of judgment documents, this paper proposes the interpretability analysis technology of judgment documents based on data quality which aims to improve data value density. It combines advanced technology like text abstract and dependency parsing with judicial rules to measure and improve the quality of judgment documents by fact model and multi-dimensional measurement system about data/information quality. The multi-dimensional measurement system covers six dimensions of standardization, completeness, readability, responsiveness, delay, and richness, with a total of 24 indexes. Judicial rules and advanced technology ensure the accuracy and utilization of judgment documents respectively.

In order to reduce the cost of users and maximize access to document information, the analysis service described in this paper is the core of the technology. Based

on the low-cost and high-efficiency operations, this paper designs a fact model and builds a multi-dimensional measurement system about data/information quality espectively from the perspective of single file and batch file. After automatically processing and analyzing data via NLP, the tasks present the interpretability analysis reports for judgment documents. Meanwhile, problem localization of documents is detected and suggestions for repair are provided. This paper designs a system, including data interaction module, measurement analysis module, and quality improvement module. It provides a user-friendly interactive interface to help users easily analyze data and perceive representative information.

keywords: Data Quality, Interpretability, Text Abstract, Quality Improvement

目　　录

目　　录 ····· **v**

插图清单 ····· **ix**

附表清单 ····· **xi**

第一章　引言 ····· **1**

1.1　课题背景和意义 ····· 1

1.2　国内外研究现状 ····· 2

1.3　研究目标与研究内容 ····· 4

1.3.1　司法文本分析技术及度量指标体系 ····· 5

1.3.2　自动化质量提升 ····· 5

1.4　本文组织结构 ····· 5

第二章　技术综述 ····· **7**

2.1　数据质量评估 ····· 7

2.2　自然语言处理技术 ····· 10

2.2.1　分词 ····· 10

2.2.2　词性标注 ····· 11

2.2.3　命名实体识别 ····· 12

2.2.4　文本摘要技术 ····· 13

2.3　文本数据可视化技术 ····· 15

2.4　数据质量提升 ····· 16

2.5　系统开发技术栈 ····· 17

2.5.1　Redis ····· 17

2.5.2　Django ····· 18

2.5.3　Vue ····· 18

2.6　本章小结 ····· 18

第三章　需求分析与概要设计 ····· **19**

3.1　裁判文书可解释性分析系统整体概述 ····· 19

3.2　裁判文书可解释性分析系统需求分析 ····· 20

3.2.1 涉众分析 …… 20
3.2.2 功能性需求 …… 21
3.2.3 非功能性需求 …… 21
3.2.4 系统用例分析 …… 23
3.3 裁判文书可解释性分析系统概要设计 …… 28
3.3.1 系统总体架构设计 …… 28
3.3.2 架构建模 …… 30
3.4 数据交互模块 …… 34
3.5 度量解析模块 …… 35
3.5.1 文书结构解析 …… 36
3.5.2 细粒度分析 …… 37
3.5.3 粗粒度分析 …… 43
3.6 质量提升模块 …… 45
3.7 数据库实体设计 …… 46
3.7.1 结构设计 …… 46
3.7.2 数据库表设计 …… 48
3.8 本章小结 …… 52
第四章 详细设计与实现 …… 53
4.1 数据交互模块 …… 53
4.1.1 详细设计 …… 53
4.1.2 关键代码 …… 54
4.2 度量解析模块 …… 55
4.2.1 详细设计 …… 55
4.2.2 关键代码 …… 59
4.3 质量提升模块 …… 61
4.3.1 详细设计 …… 61
4.3.2 关键代码 …… 63
4.4 本章小结 …… 64
第五章 系统测试与分析 …… 65
5.1 功能测试 …… 65
5.1.1 测试设计 …… 65
5.2 接口测试 …… 68

5.2.1 测试设计 …… 69
5.2.2 测试结果 …… 70
5.3 本章小结 …… 71
第六章 总结与展望 …… **73**
6.1 总结 …… 73
6.2 展望 …… 74
致　谢 …… **75**
参考文献 …… **77**
简历与科研成果 …… **83**

插图清单

2-1 SummaRuNNer 模型 …… 14

3-1 系统用例图 …… 24
3-2 系统整体架构图 …… 29
3-3 逻辑视图 …… 30
3-4 开发视图 …… 32
3-5 过程视图 …… 33
3-6 物理视图 …… 34
3-7 数据交互模块设计类图 …… 35
3-8 度量解析模块设计类图 …… 36
3-9 质量提升模块设计类图 …… 45
3-10 数据库 E-R 图 …… 47

4-1 数据交互模块顺序图 …… 53
4-2 数据主界面及上传界面 …… 54
4-3 度量解析模块顺序图 …… 55
4-4 任务管理界面 …… 56
4-5 细粒度分析报告（一） …… 56
4-6 细粒度分析报告（二） …… 57
4-7 粗粒度分析报告（一） …… 57
4-8 粗粒度分析报告（二） …… 58
4-9 细粒度分析任务关键代码 …… 60
4-10 文本摘要关键代码 …… 60
4-11 质量提升模块顺序图 …… 61
4-12 问题文书管理界面 …… 62
4-13 文书问题列表 …… 62
4-14 问题文书检测关键代码 …… 63

5-1 线程组配置示例图 ····· 70
5-2 HTTP 请求配置示例图 ····· 70
5-3 启动细粒度分析任务测试结果 ····· 71

附表清单

2-1 质量维度分类 …… 8
2-2 质量维度适用的数据类型 …… 9
2-3 Flesch-Kincaid 可读性测试对应分数表 …… 9
2-4 信息质量活动的定义 …… 16

3-1 涉众分析 …… 20
3-2 系统功能性需求列表 …… 22
3-3 系统用例列表 …… 23
3-4 数据上传交互用例描述 …… 25
3-5 文件管理用例描述 …… 25
3-6 启动分析任务用例描述 …… 26
3-7 分析报告交互用例描述 …… 26
3-8 问题文书管理用例描述 …… 27
3-9 问题文书修复用例描述 …… 28
3-10 民事裁判文书结构 …… 38
3-11 规范性指标 …… 40
3-12 最佳审判期限（分阶段） …… 42
3-13 问题参考列表 …… 46
3-14 user 表 …… 47
3-15 file_pre 表 …… 48
3-16 upload_record 表 …… 49
3-17 task 表 …… 49
3-18 report 表 …… 50
3-19 report_single 表 …… 50
3-20 report_batch 表 …… 51
3-21 problem_file 表 …… 51

5-1 数据上传交互测试用例 …… 65

5-2 文件管理测试用例 ····· 66
5-3 启动细/粗分析任务测试用例 ····· 66
5-4 分析报告交互测试用例 ····· 67
5-5 问题文书管理测试用例 ····· 67
5-6 问题文书修复测试用例 ····· 68
5-7 接口测试用例表 ····· 69
5-8 系统任务响应时间表 ····· 71

第一章　引言

1.1　课题背景和意义

2020 年的《最高人民法院工作报告》指出，面对全球范围内案件尤其民商事案件持续较快增长的趋势，中国法院必须靠深化司法改革、建设智慧法院，加快推进审判体系和审判能力现代化，不断提高司法质量、效率和公信力 [1]。

司法数据作为智慧法院建设中不可缺少的一环，扮演着重要角色。截止至 2020 年 12 月，中国裁判文书网的公开文书已超过 1 亿篇；据同年 4 月截止的数据统计，中国审判流程信息公开网公开案件两千九百万件，包括其他平台在内总信息达 15 亿项。在这个庞大的数字下，司法的数据质量得到了广泛关注。

裁判文书作为法院的“司法产品”，是法院履行审判职责、展示司法正义的载体，在解决社会纠纷、塑造良好社会秩序上的基础功能和重要使命更为突出。在裁判文书公开的大背景下，裁判文书的受众不仅仅只有当事人及案件相关者，还延展到中国裁判文书网的潜在浏览者、法律实证研究者（如检察官、律师、高校老师等）。面对海量的、动态的、多样的裁判文书时，无论是仅获取文书信息或进一步利用该信息达到其他目的，难免需要耗费较多时间阅读。

裁判文书的质量在一定程度上是正义的表征，可简单且直观地传达给公众。因为裁判文书来源于法官，文书的质量控制可从侧面反映法官的专业能力和办案水平，且关系着法院的整体形象和司法权威。由于不同地域、不同级别的人民法院的个体差异，以及法官业务能力和审判质效的不同，导致文书质量参差不齐。在人民法院系统的文书上网公布过程中，各法院的文书上网公布工作和情况由其上级指导和检查、考核。法院负担本就繁重，案多人少，且我国的民商事案件呈爆发式增长的趋势，仅依靠人力层层审批远远不够。

司法领域智慧法院建设广泛使用的深度学习技术依赖于大批量的司法数据。尽管公开的裁判文书数量庞大，但由于我国非判例法国家，部分案例显得数据量有限。依赖原始数据的数据扩增技术出现可使模型具有更好的泛化能力。另一方面，即使在具备足够多的数据的情况下，数据增强可以根据需要提升数据质量防止获得不理想的模型，预防出现包括但不限于过拟合等现象。然

而这些用于数据增强的扩增数据需要通过数据质量度量及分析确认其有效性。

充分挖掘司法数据可在民生层面上发挥巨大价值。政府在履行其社会职能，提高社会福利的整体水平的过程中可以充分利用全国或地方的海量文书数据。国家或地方政府从大量案件中了解地域、行业或者审级的裁判分布，预测某类纠纷的增长、爆发及特定行业的法律风险点分布，分析当地潜在的社会问题，及时作出应对策略，履行社会职能，为官方立法修法提供数据支持。司法数据分析依赖技术的实现，由于政府工作人员非专业技术人员，需要借助工具提高科学决策化水平。

基于上述背景，本系统面向多类用户解决文书篇幅冗长、文书质量问题、文书数量众多、文书集有效性验证、文书集信息提取等多个问题，在用户提供文书数据和少量配置的情况下，自动获取文书或文书集的核心信息，对数据进行多维评估，同时检测文书问题定位和提供修复建议，从而帮助用户快速感知文书数据传达的信息并控制文书质量。

1.2 国内外研究现状

20 世纪 90 年代，众多专家投身数据质量领域研究，并提出了许多不同的数据质量定义以及多套质量评估体系。尽管数据质量在多个文献中的定义不尽相同，但是都具有一个共同的特点，即数据质量的好坏除了考虑自身特性，还要考虑数据的使用场景和业务背景，涉及到的具体的业务流程和相关用户。麻省理工大学教授 Richard Y. Wang 带领的 MITIQ 小组的研究在数据质量领域具有深远影响，他们将“数据质量维度”定义为一组数据质量属性 [2]，包含 15 个数据质量维度的 4 个类别。

相比于国外对数据质量的研究历史，我国起步较晚 [3]。解放军总参谋部第 63 研究所于 2008 年校正了对数据质量定义、问题来源、提高途径等基本问题的认识偏差 [4]。计算机网络信息中心提出了数据质量评估方法和指标体系，将其分为外部形式质量，内容质量和效用质量三大类，各个类别包含质量特征与评估指标。目前更高层面的数据质量评估框架建设已有方法包括三种：层次分析法 [5]、模糊综合评估法 [6] 和云模型评估法 [7]。现阶段我国法院的数据质量检测方法使用层次分析法层层递进构建数据质量框架 [8]，然而该质量框架的指标仅能检测结构化文本的合规性，存在诸多弊端。

司法领域的数据质量研究一直广受重视。最高人民法院在 2015 年度裁判文

书点评会议[①]提出“要全面深化裁判文书质量管理，大力提升裁判文书水平”。近年来，全国各大法院纷纷出台关于裁判文书数据质量管理的文件，要求促进创建新型裁判文书质量评查评价体系，并且具有科学合理、可操作性等特性。徐州市中级人民法院提出了文书量化评估模型[②]，选取释法说理和行文规范两个角度展开。以裁判文书质量为自变量 y，以行文规范和释法说理两项为基本因变量 X_1、X_2 设定指数模型，即 $y = f(0.2X_1 + 0.8X_2)$，y 的分值越高，代表文书行文规范性程度越高、文书裁判说理质量整体较优。天津市滨海新区人民法院分析裁判文书存在的问题，明确裁判文书技术、内容、制作、评查等标准[③]。在司法数据流动层面，高杰提出从数据确权、数据采集、使用场景、数据安全、数据价值等方面开展对智慧云法院司法数据的综合治理研究 [9]。

“建设智慧法院”是国家信息化发展战略之一，该观点最早于 2016 年提出。我国各地法院积极响应号召，都在探索某种形式的立足于时代发展前沿的智慧法院建设 [10]。近年来，中国对司法公开理念的大力贯彻推动了中国裁判文书网的良好发展，使得裁判文书拥有了多个适用于科学研究的优点，如海量性、丰富细致性、不反应性等 [11]。然而如何利用这些数据，获取信息、发挥价值、反哺司法工作以推动智慧法院建设，成为司法领域的下一步工作。由国家司法领域相关单位部门指导举办的“中国法研杯[④]”至今已举办三届，针对在大数据与信息时代下如何融合司法大数据与自然语言处理技术展开研究，促进智慧司法相关技术的发展。大赛聚焦多个现实世界的任务，提供大量处理标记过的高可用文书数据，邀请学术界和工业界的研究者和开发者参与，在司法工作存在的许多问题研究中已取得较大突破。目前应用于司法领域的人工智能技术层出不穷，较为成熟且应用广泛的包括案例筛选 [12][13]、罪名预测 [14]、法条推荐 [15]、刑罚预测 [16]、庭审语音同步转录等。与裁判文书相关的的热门研究包括阅读理解 [17]，构建的模型通过对文书的读取分析推理，回答针对文书关于如时间、地点、人物关系的提问；要素识别使用模型通过对文本中每个句子的识别，寻找其中的关键案情要素；相似案例匹配是对文书之间的相似度进行计算，从候选集文书中找到与询问文书最为相似的一篇文书作为匹配；论辩挖掘指寻找由于起诉方和应诉方观点不同形成的庭审过程中双方的争议焦点，争议焦点是整场庭审的关键；司法摘要则是对判决书的内容进行压缩、概括和总结，但是仍然可以反映案件审理过程中的各项重要信息。

[①] http://www.court.gov.cn/zixun-xiangqing-16515.html
[②] http://xzzy.chinacourt.gov.cn/article/detail/2019/07/id/4145669.shtml
[③] http://bhxqfy.chinacourt.gov.cn/article/detail/2014/04/id/1283763.shtml
[④] http://cail.cipsc.org.cn/index.html

由于裁判文书包含信息量大、内容复杂、篇幅冗长，且大数据时代下文书量级较大，阅读者面对大量文书，如何快速知悉其核心信息成为一大难题。基于裁判文书内容以及裁判文书集，朱箭飞曾开发了“律师之家”和 openLaw[①]两款工具，帮助律师、法官、司法行政人员、法学生等多种用户类型的裁判文书使用者分析文书和文书集。前者实现对裁判文书的自动分段，提高阅读裁判文书的效率，让用户可以更加容易地找到所需要的信息。后者是裁判文书检索网站，并且具有初步的统计分析功能。林学飞认为，文书摘要在飞速增加的案例数量面前是高效编写和发现指导性案例的捷径 [18]，因此将文本摘要技术应用于司法领域具有可行性，可快速了解文书信息，发挥出较大的价值。在数据集的分析上，Cristian Felix 等人 [19] 开发的 texttile 是一款适合同时展示分析结构化数据和非结构化文本的数据可视化工具。基于上述目的和现有研究，本文提出了数据质量驱动的裁判文书可解释性分析技术，结合数据质量、自然语言处理、文本摘要、可视化等多项技术，聚合生成可解释性分析报告，帮助用户阅读了解评估文书及文书集。

1.3 研究目标与研究内容

本文的工作重点在于针对裁判文书篇幅过长，裁判文书集海量数据信息繁杂冗余等问题，提出数据质量驱动的裁判文书可解释性分析技术。同时为减少文书数据使用者的工作量，自动检测文书存在问题并进行质量提升。裁判文书是司法审判活动的产物，案情的载体，根据司法规则解析裁判文书保证了其准确性，根据互联网技术解析裁判文书提高了其利用率。另外，本文的“可解释性”不作为数据质量的一个衡量指标，而是作为文书数据的一种性质，一种技术手段的命名。

为降低裁判文书使用者的时间成本，且在最大化获取文书信息的同时减少为提高数据质量而进行数据处理的工作量，本文以分析服务作为技术核心，在低成本高效率的操作基础上，分别从细粒度和粗粒度层面（二者分别面向单篇文书和文书集）设计事实模型和构建基于数据/信息质量的多维度量体系。利用自然语言处理技术自动化处理分析数据，呈现文书及文书集的数据质量驱动的可解释性分析报告，同时检测文书问题定位，并提供修复建议。本文主要对司法文本分析技术及度量指标体系和自动化质量提升两部分展开研究。

[①]http://openlaw.cn/

1.3.1 司法文本分析技术及度量指标体系

本部分旨在协助裁判文书的制作者、审核者、阅读者、研究者快速知悉文章主题和核心内容，了解文章脉络和阅读线索，辅助决策支持。实质为将数据转化为信息，再将信息转化为决策。

裁判文书是从法官或法官助手等司法工作人员视角撰写的，而裁判文书的受众甚广，既包含具有行业知识的研究员，也包含有限阅读水平的普通人民群众。本技术以普通群众角度重新解构裁判文书信息，依据国家标准和文书规范总结归纳民事裁判文书结构，精炼提取各部分内容信息，摘要长文本段落描述，构建度量体系对评估文书多个维度质量。在文书集层面，客观展示文书集数据内容信息，构建数据集度量体系，协助数据人员判断数据集的数据质量，包括数据是否异常、是否具有代表性。

基于上述分析和度量指标的计算结果聚合生成可解释性分析报告。数据可视化研究已经发展甚久，然而实际情况中，单纯可视化过于抽象，无法直观清晰表达文本信息。相比于数据可视化，可视化与文本摘要相结合对于用户更加友好。提取文本关键信息，与数据质量相结合，将数据指标以文字、数字以及简易图像相结合，大大增加了数据可读性，我们将之称为文本可解释性分析。

1.3.2 自动化质量提升

Jim Barker 将数据质量问题分为两类，第一类为通过检测数据完整性、一致性、唯一性、有效性等数据质量维度判定，这类数据由于常识或规定等已知信息可以定位。第二类为藏于细节中的问题，往往需要多方的输入确认。尽管如此，第一类问题涵盖了 80% 的数据质量问题。针对第一类质量问题，本文结合司法领域相关知识，根据已有裁判文书数据统计分析获得预设问题，从而构建问题检测体系。在重新解构及度量解析裁判文书的过程中，系统检测异常问题并记录定位。为保障问题定位的有效性，应考虑将筛选后的数据待用户二次确认再完成数据质量提升。

1.4 本文组织结构

在国家大力建设“智慧法院”背景下应使用先进技术充分利用裁判文书数据，高效分析文书信息，最大化发挥数据信息价值。据此，本文将为裁判文书

建立全面的多维度量体系，主要从事实和数据/信息质量多个维度分析，从文书粒度和文书集粒度两个层面进行评估，并以此衍生为可用的分析系统。首先分析了可解释性分析裁判文书及裁判文书集数据的必要性及意义。随后介绍本技术所涉及的相关技术。在涉众分析的基础上完成系统的需求分析（包括功能性与非功能性需求），并据此完成了概要设计。从多个视角解读系统结构，将系统解耦为多个模块并对模块的具体实现进行了介绍。最后对系统进行了测试与分析以及总结，对系统不足及改进方向加以说明，提出了未来展望。本文组织结构如下：

第一章为引言部分，本部分介绍了项目的司法、数据、人工智能技术交叉学科背景，及其国内外研究现状，并讲述了本文的研究目标，对司法文本分析技术及度量指标体系、自动化质量提升两部分研究内容进行简要介绍。

第二章为技术综述，本部分介绍了本技术关于数据质量、自然语言处理、质量提升、系统实现的核心概念和技术方案。

第三章为系统的需求分析与概要设计，本部分针对系统涉众分析、功能性和非功能性需求分析、用例分析设计了系统整体架构，通过 4+1 视图及多个模块解析设计类图展示整个系统，并在核心模块建立了多维度量体系。

第四章为系统的详细设计与实现，本部分根据第三章系统模块设计，利用顺序图、关键代码展示数据交互模块、度量解析模块和质量提升模块的实现细节，并进行了阐述。

第五章为系统测试与分析，本部分使用测试用例对系统进行功能测试和接口测试，确认系统是否满足功能性需求和非功能性需求。

第六章为总结与展望，本部分对论文所完成的工作进行了总结，并对系统研究和开发中存在的不足与优化方向进行了探讨。

第二章　技术综述

2.1　数据质量评估

影响数据质量定义的因素有许多，随着时间的推移或是应用领域的不同，以及专家学者思想认识的差异，其定义在不断变化。因此，数据或信息质量的概念取决于数据的实际使用。Alizamini[20] 等人认为通过数据质量管理可以使数据达到使用者的需求或在使用时发挥其价值,这也是评估数据质量的目的。在评估或改善数据质量的层面上，该领域的研究者们的一个统一认知是根据信息或数据的特征对其进行分类，数据质量维度的概念应运而生。据此，通过分解数据质量维度主要步骤所设计的测量和管理数据质量及信息的方法层出不穷。

为了便于管理数据的维度和分类，数据科学家们开发了工具在数据创建或使用过程中对其进行质量控制，同时在单独使用部分数据方面也更加便捷。数据质量的各个维度在描述数据某个角度的特征上都具有较强的代表性。实际上，数据质量维度偏向于数据值的质量度量，即数据的内涵，又被称为数据的模式质量。早期数据质量领域的专家以定性方式定义数据维度及模式维度，然而对于这些定义并没有进一步提供定量度量指标。

在追求高数据质量的目标中，首先必须了解质量的含义和测量方法。经常提到的数据质量维度是准确性，完整性，一致性和及时性。这些维度的选择主要基于直觉理解 [21]，行业经验 [22] 或文献综述 [23]。

Wang[24] 在本体论基础上严格定义数据质量维度，该方法以面向系统设计的术语代替“数据为中心”思想分析数据质量，进而反映信息的预期用途。由于信息系统服务于用户，因而用户的观点是定义数据质量的标准。他们总结了最常引用的 26 个质量维度，将这些维度根据内部视图和外部视图又分别归结为数据相关类和系统相关类。在分析了定义为内部视图的五个维度（包括准确性与精确性、可靠性等）后，生成了四个固有的（面向系统的）数据质量维度。

表 2-1: 质量维度分类

质量维度	特征
Accuracy	准确性、正确性、有效性和精确性是衡量表达现实的真值的能力。
Completeness	完整性、针对性和相关性是表达事物被关注方面完备程度的能力。
Consistency	一致性、内聚性，描述的是数据信息表示与真实世界情况的符合程度。
Redundancy	冗余性、简约性、紧凑性和简洁性是指通过最少的信息资源表示现实的能力。
Readability	可读性、可理解性，清晰度和简洁性是信息易于理解和实现的程度。
Accessibility	可访问性，描述的是数据在被用户访问获取时的一种性质，涉及多种因素，除技术能力外还与用户的自身能力有关。
Trust	可信度，可靠性和声誉，重点关注从权威来源获得的信息量。
Usefulness	有用性代表的是用户获取的信息中有效信息量占比程度。

根据 [25] 的维度聚类，即通过计算维度相似性分类集群，信息质量维度可总结为一个用于比较不同信息类型维度的框架 [26]。表 2-1展示了各个分类信息。每种维度适用的数据类型有所不同，如表 2-2所示，其中 Linked data 是互联网上相关联的数据集合。语义网是一个由数据组成的基于关联数据规则的网络，用户如共享文档般共享数据，可自由创建应用。为了使数据网络成为现实，让网络上的大量数据以一种标准的格式可用是很重要的，这种格式可以通过语义网工具访问和管理。此外，语义网不仅需要访问数据，而且数据之间的关系也应该可用，以创建一个数据网，而不是纯粹的数据集合。

当处理结构化数据时，考虑的是数据质量；当考虑根据其他数据模型表示的信息时，则指的是信息质量。裁判文书是文本数据，司法内部使用 xml 格式存储文书文件，介于结构化数据与非结构化数据之间。xml 是一种可扩展标记

性语言，它具有结构化特点，但又由于具有可伸缩性而无法完全结构化。裁判文书以 xml 格式存储时，描述性内容为文本——非结构化数据。因此分析裁判文书需要从结构化与非结构化两种数据类型的角度分别评估，构建度量体系。

表 2-2: 质量维度适用的数据类型

质量维度	数据类型
Accuracy	结构化数据
Completeness	结构化数据
Consistency	结构化数据
Redundancy	Linked data
Readability	文本-非结构化数据
Accessibility	网站数据
Trust	网络数据源
Usefulness	图片

文本的可读性是一个重要的评估维度。可读性属于应用语言学范畴，相关研究在美国发端,已有 90 多年的历史。英语可读性研究在中国发展已 20 余年，现有的大部分成果虽然在某些步骤或关键点具有创新，然而阅读教学评价仍沿用国外的部分公式。关于中文可读性研究成果更是稀少，尤其在司法领域。

表 2-3: Flesch-Kincaid 可读性测试对应分数表

得分	学习水平	备注
100.00-90.00	五年级	非常容易阅读，普通的 11 岁学生可轻松理解
90.00-80.00	六年级	易于阅读，面向消费者的会话英语
80.00-70.00	七年级	相当容易阅读
70.00-60.00	八年级和九年级	简单易懂，13 至 15 岁的学生都能理解
60.00-50.00	十到十二年级	相当难以阅读
50.00-30.00	大学	难以阅读
30.00-10.00	大学毕业生	很难读懂，大学毕业生能理解
10.00-0.00	专家	极难阅读

Flesch-Kincaid 可读性测试公式 [27] 被认为是最古老和最准确的可读性公式之一，以词长和句长作为自变量，通过建立多元线性回归公式来评估文本难

度。RE 是一个 0 到 100 之间的数字，数字越大文本就越容易阅读。适合阅读人群对应分数如表 2.3 所示。它是一个评估读者等级水平的简单方法，也是为数不多的我们可以依赖而不需要太多审查的精确测量方法之一。该公式被微软 Ofice Word 应用于评估文本难度 [28]。

$$RE = 206.835 - 1.015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right) \tag{2-1}$$

The Gunning-Fox 是经典的可读性评估算法，也是一个评级公式。其根据句子长度和单词复杂度判断阅读文本的难度，并且后者与前两个因素呈正比关系 [26]。Gunning fog 指数通常用于确认文本是否能被预期受众轻松阅读。供广大读者阅读的文本结果值一般需要小于 12。需要接近普遍理解的文本的结果值一般需要小于 8。

$$0.4 \times \left[\left(\frac{\text{words}}{\text{sentence}}\right) + 100 \times \left(\frac{\text{complexwords}}{\text{words}}\right)\right] \tag{2-2}$$

ARI 为另一种可读性度量算法，该算法通过计算获得自动可读性指数，度量公式如下所示 [29]。

$$ARI = 4.17 \times \frac{\text{characters}}{\text{words}} + 0.5 \times \frac{\text{complexwords}}{\text{sentences}} - 21.43 \tag{2-3}$$

吴思远等人于 2020 年提出了预测汉语文本可读性的多层面、多维度特征体系 [30]，从汉字、词汇、句法和篇章四个层面出发，包含 13 个维度共 104 项指标。并收集了人多个出版社的现行语文教科书，通过 OCR 和人工校对的方法，建立了语文教材语料库用以验证中文文本可读性特征体系的作用，在模型预测准确率方面仍有待优化。

2.2 自然语言处理技术

2.2.1 分词

分词是中文自然语言处理的基础，以词汇为单位分解句子或段落、文章，进而协助完成语言的多角度分析。目前中文自然语言处理的研究有了一定的成就，多款工具都可以对文本进行分词，并达到不错的效果，如 jieba、哈工大的

pyltp、中科院的 NLPIR、清华大学的 THULAC 等。这些分词工具主要基于词典、统计机器学习或深度学习实现，下述为几种实现方法的介绍。

依赖词典的分词工具，其原理是字符串使用前序准备的词典与其词汇相匹配。匹配过程中的方向维度包括正向与逆向，长度优先维度包括最大和最小。两个维度可组合出多种方法。其中，统计结果表明在最大匹配法中，逆向法相比于正向法精确度更高。而实际使用时，双向匹配分词即两个相结合的分词方案可以提高系统分词的准确性。首先对文本进行粗切分，将分解后的句子使用正/逆向最大匹配法分别进一步扫描识别。如果两者的分词结果相同判为成功，反之取它们的最小集。依赖词典分词速度快且直观，然而没有考虑文本上下文语义特征，在未登陆词存在的情况下分词效果差。为了更好地分词，许多工具包都提供了调整词典、添加词典的功能。

基于统计机器学习常用的算法是 SVM、CRF、HMM 等。工具 pyltp 底层原理就是 CRF，即条件随机场。对汉字标注训练，同时考虑词频与上下文语义关系，给标注序列打分，利用模型不错的学习能力，选出靠谱的标注序列，可有效识别部分歧义词和非字典词汇。pyltp 的一大特点是支持个性化分词。个性化分词的目的是为跨领域研究及数据使用提供便捷的解决办法，减少工作量，提高工作效率。当用户需要切换至新领域时，只需要标注少量数据，就可以在原有领域数据基础上完成增量训练。在充分利用了原有领域的丰富数据的同时考虑了目标领域的特性。

基于神经网络的分词器是深度学习的一种。其算法 [31] 为双向 LSTM+CRF，本质为序列标注，适用于命名实体识别，且分词准确率极高。

中文分词的难点主要有三类。一是不同工具分词标准不统一，如姓名的姓与名是否分开，如日期的年月日是否分开，均需根据不同需求制定相应的分词标准；二是歧义，同一字符串出现多种分词情况，由于分词粒度粗细差异产生的组合型歧义，由于某个字符与其前和后字符均可组成词汇的交集型歧义，由于未考虑上下文语义环境仅根据句子本身语法语义切分即使采用人工切分也存在的真歧义；三是新词，某些词汇未被词典收录且依赖人类认识不断更新，包括机构名、产品名、商标名等。

2.2.2 词性标注

词性标注是基于分词的基础上的进一步研究工作，用单独的标签标记每个词。词性是词汇基本的语法范畴，它代表词汇在句子中的角色，例如名词、代

词、动词等。目前已有许多工具支持词性标注，然而这些工具间的标注存在一定的差异。HandLP 使用的是一阶隐马尔可夫模型，由于训练数据涵盖了 2014 年和 98 年的人民日报的语料及特有词汇，兼容了两个标注集标准。但是这些训练数据存在词性单一问题，可能有部分错误存在。jieba 采用基于统计模型的标注方法，是和 ICTCLAS 兼容的标记法。ICTCLAS 现在已经更新为 NLPIR。pyltp 使用的是国标 863 词性标注集。

目前关于词性标注的研究较多，较为常见的是基于规则、统计、规则与统计结合、深度学习的词性标注方法四类。规则类标注方法 [32] 是最早使用的方法，通过上下文语境获取规则进行词性标注。统计模型类标注方法研究包括了 HMM[33]、MEMM[34][35]、CRF[36] 等多种统计模型，通过学习条件概率不断提高标注任务的成绩。结合规则与统计模型的方法 [37] 是在使用统计模型的基础上，对可疑的统计结果即兼类词，结合规则消解歧义确定词性。基于深度学习的词性标注方法 [38][39] 利用神经网络的损失函数提高词频低的词汇判断或者使用新型深度语境化词表征对兼类词建模以提高准确度。

中文词汇标注研究中存在着几大难题。根据汉语言的特性，无法通过词的形态改变判断词性。博大精深的中国文化创造了某些兼具多种词性的词汇，且这类词汇越常见兼类现象越严重。尽管这类词汇仅占汉语词汇的一小部分，但使用程度高且覆盖许多词类，词性容易出错。以及由于主观因素如词性划分的目的、标准等导致的歧义，词类划分的粒度和标记的符号存在差异。

2.2.3 命名实体识别

命名实体识别的任务是识别如姓名、地点、机构名称或者专有名词等在文本中具有特定意义的实体，这些藏于待处理文本的实体可分为三大类和七小类。通过识别文本中的实体，或者时间，数字等信息，可以促进如信息抽取、文本问答等技术的发展。

命名实体识别的难点在于形式多样，构成复杂，边界模糊。主要的实现方法分为基于规则和词典、统计、以及前两者混合三类方法。以上方法在不同程度上都需要依赖于语料库。命名实体的众多技术的本质还是序列标注任务。

目前开源的中文命名实体工具并不多，明确有 NER 标记的包括斯坦福大学的 NLP 组的 Stanza，百度的 Paddle Lac，哈工大的 LTP，而其他这些测试过的开源 NLP 基础工具，需要从词性标注结果中提取相对应的专有名词，也算是一种折中方案。

2.2.4 文本摘要技术

文本摘要的本质是信息过滤。输入输出均为文本，经过算法等技术流程处理后在量级上发生差异，将较长文本转化为简短、流畅且准确的摘要，保留文本识别处理前的传达的重要事实和信息，从而增强文本的可读性。

文本摘要的实现方法从广义上可总结为两类：抽取式和生成式。

抽取式摘要可简要理解为从原文中抽取关键词或关键句组成概要，可细分为传统抽取式摘要方法和基于神经网络的抽取式摘要方法。

传统抽取式摘要方法主要使用图方法、聚类等方式完成无监督摘要，比较经典的包括抽取文章的前三句作为文章摘要的 Lead_3；TextRank[40] 和 LexRank [41] 使用 PageRank 算法思想将文本的每个句子作为节点，计算句子间相似度构造无向有权边从而构建图模型，不断迭代计算节点得分直到所有得分不再变化时迭代停止，最后选取 N 个最高分节点作为摘要; 聚类方法将句子作为最小单位编码 [42]，应用 K 均值和 Mean-Shift 执行句子聚类，选取 N 个类别中离质心最近的 N 个句子获得摘要。

基于神经网络的抽取式摘要方法是提取训练数据的特征向量，构造机器学习模型的有监督学习方法。主要包括如下所述的序列标注方式、Seq2Seq 方式、句子排序方式。

将文本建模为序列标注的二分类任务，每个句子都有一个二分类标签，任务输出的分类标签为 1 的句子组成摘要。Nallapati 等人 [43] 构建的模型 SummaRuNNer 使用双向 GRU 分别建模词语级别和句子级别的表示。如图 2-1所示，*WordLayer* 代表词语层，*SentenceLayer* 代表句子层。通过输出的每个句子二分类标签，判断其是否为最终摘要的组成部分。Zhang 等人 [44] 提出序列标注结合 Seq2Seq 学习句子压缩模型进而衡量句子选择好坏，再结合强化学习完成 Latent 模型训练。Latent 模型将序列标注作为中间的步骤得到概率分布后从中采样候选摘要集合，利用标准摘要中的信息计算损失。

Seq2Seq 方式同时考虑了句子和关键词，SWAP-NET 模型 [45] 直接使用 Seq2Seq 模型交替生成词语和句子的索引序列，解码过程中不断计算 Switch 概率指示生成词语或者句子，获得词语和句子的混合序列。根据句子概率及得分从产生的句子集合中选择前 N 句作为摘要。句子排序方式相比于序列标注任务针对每个句子的输出，使用的是属于摘要句的概率，而不是二分类标签。较为经典的算法是 TextTeaser，根据句子长度距离理想长度远近、句子在全文的位置（段落的首句为核心句概率大致为 70%）、句子关键词词频分布进行打分，

对于上述几个统计指标的打分累加，同时考虑摘要可读性，倒排句子重要性得分且按在文中出现顺序排序。其中，使用算法 MMR[46] 加入惩罚机制可以解决 TextTeaser 结果中得分较高的 k 个句子语义相近问题。

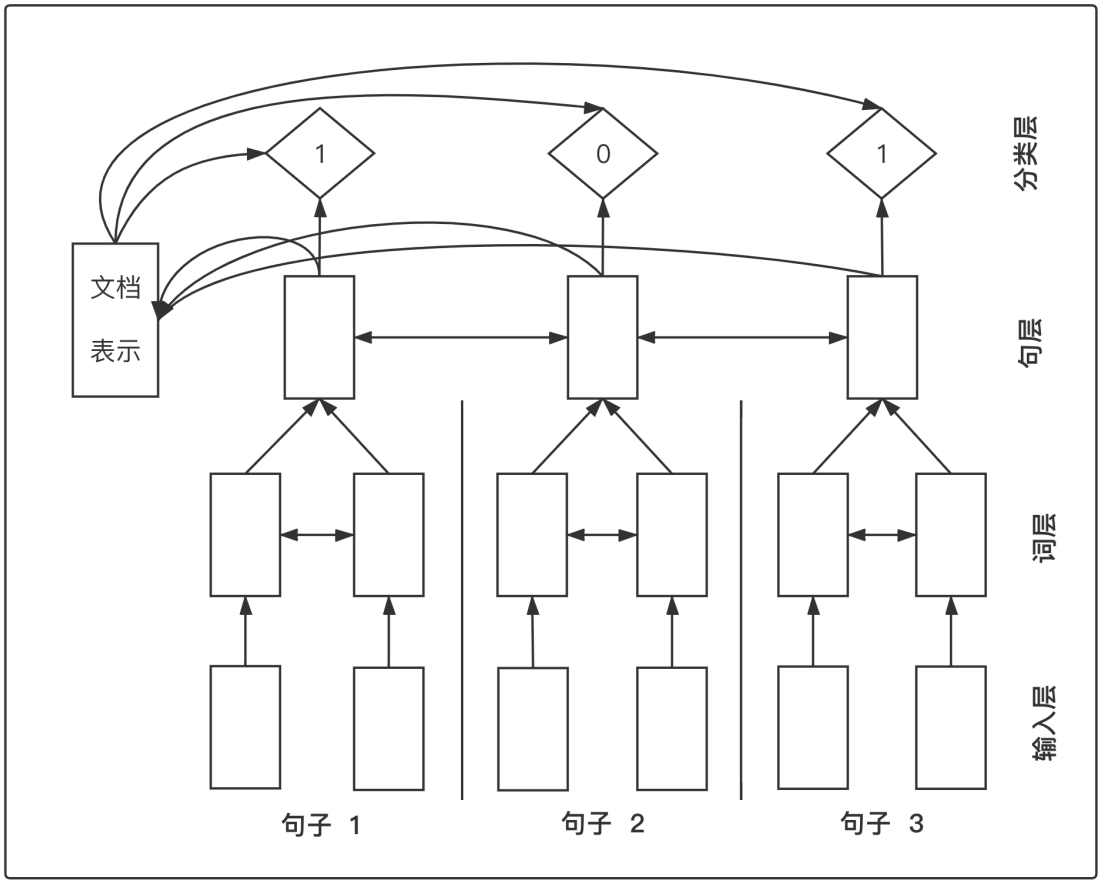

图 2–1: SummaRuNNer 模型

由于抽取式摘要方法存在灵活性差、连贯性差等问题和深度学习的极速发展，生成式摘要方法应运而生。生成式摘要的实现过程为使用自然语言理解进行篇章理解，再利用自然语言总结上下文生成摘要，允许摘要中包含新词汇或短语，灵活性高。然而此类方法存在三大难点，理解文本所表达的意思和话题，生成连贯可读性强的新文本，精炼总结文本核心思想。

早期的研究如基于 RNN 的 Attention Model[47]、基于 CNN 的 ABS[48] 应用机器翻译（Machine Translation）中的 Encoder-Decoder 框架和 Attention 机制在一定程度上实现了自动文本摘要，然而效果一般。在生成式领域较为受欢迎的框架 [49] 在基于注意力机制的 Seq2Seq 基础上增加了 Copy 和 Coverage 机

制，分别解决了仅前者带来的未登陆词问题以及生成重复问题。

抽取生成式摘要结合了抽取式摘要和生成式摘要二者的优点，在一定程度上可以扬长避短，专家学者逐步往这个方向开展研究。摘要任务可分为两个步骤，定位重要信息，改写文本内容。“Bottom Up”方式 [49] 首先使用“content selector”选择关键内容，建模为词语级别序列标注任务，获得词汇粒度的标签；继而利用 pointer-generator 网络生成摘要，利用选择内容部分计算的概率改进原有的 attention 概率。Li 等人 [50] 使用 TextRank 算法代替序列标注方式获得关键词，并通过神经网络表示，后续同样使用 pointer-generator 网络生成摘要。

当前关于文本摘要的评价方法除人工评价外，公认的评价方法只有 ROUGE 指标 [51]，思路为使用待测摘要与参考摘要的 n 元组共现统计量，通过一系列标准如 ROUGE-N、ROUGE-L、ROUGE-W、ROUGE-S 和 ROUGE-SU 进行打分，即使用共同出现次数、最长相同文本长度、带权最长相同文本长度等构造定量化指标描述待测摘要与参考摘要的相似性。

2.3 文本数据可视化技术

文本可解释性分析技术包括了依靠指标衡量文本多维特性以及文本可视化等方法，文本可解释性分析的核心是引导用户快速获取文本传达的信息。

文本可视化技术运用了数据可视化、文本分析、数据挖掘等学科的理论方法，转化复杂的文本内容，发现内在规律特征，方便发现模式，完成分析和推理。现代可视化源于 1987 年，首次科学可视化的会议由美国国家科学基金会召开，从此出现了“科学可视化”的概念。由于各个领域（如金融、医学等）的数据在时代变化下愈发复杂抽象，可视化需求逐渐变多。因此基于统计图形学的，针对抽象信息的视觉表达手段不断发展。

数据可视化包括科学可视化、信息可视化以及可视分析学。科学可视化是对物理世界的客观描述，它可以清晰直观展示数据的真实物理状态。信息可视化适用于抽象数据，如多维度数据、文本数据。为充分利用数据，创造更高价值，人们通常会对数据进行数据分析，可视分析学是该需求下的产物。可视化的设计标准首先是保证展现数据准确性，再者为有效且精准地呈现数据，在此基础上呈现数据的美观性。可视化是一种代替大段文字描述数据信息的方法，可视化的形式协助分析者直观洞悉数据所暗含的信息，从而转化为知识。

文本数据可视化属于信息可视化，分为静态文本和动态文本两种类型。其

过程为通过对原始文本的预处理、特征的提取、特征的度量等信息挖掘手段，完成图元的设计与布局，将复杂的数据转化为图形，直观地展示文本的词频与重要度、逻辑结构、主题聚类、动态演化规律等文本单元信息，语言结构信息，文本内容含义。

2.4 数据质量提升

表 2-4: 信息质量活动的定义

信息质量活动	定义
新信息获取	通过获取新的高质量数据更新原有的问题数据的一种信息获取过程。
标准化	按照已定义的标准或基准格式修改信息。
对象识别	在给定一个或多个信息源的情况下，识别出在这些信息源中能代表相同现实世界对象的记录。当信息源只有一个时，这个活动也被称为去重。
数据集成	给数据提供一个统一的视角，而这些数据是在不同且分散的数据源中的。
信息源可信度	在开放或合作性的环境中且该环境不存在或存在很少对信息质量的控制，以数据源向其他信息源所提供的信息质量为基础，对数据源评分。
质量复合	使用复合信息质量维度的代数；例如，给定两个完备度已知的数据源和一个运算符 (如: 并)，则从作为操作数的数据源的完备度开始，计算并集的完备度。
错误定位	给定一个或多个信息源以及一组关于它们的特定语义规则的情况下，找到不符合规则的记录。
错误修正	给定一个或多个信息源、一组语义规则以及一组在记录中已经识别出的错误集合的情况下，修正记录中的错误值，以此保证所有规则得以遵守。
成本优化	在众多不同成本和不同质量维度测量标准的数据源提供者中，依据最优的成本/质量比找到消费者需要的。

信息质量涉及了众多领域和维度，通过信息质量活动可以改善数据质量。在早期未曾统一定义时，许多算法、或者基于知识的技术和启发式方法被归为信息质量活动，但是只有少部分被认可。Batini 等 [26] 归纳总结了此类活动及其定义，见表 2-4。

目前，数据驱动和过程驱动是两种提高数据质量的策略，它们采用的是不同的技术 [52]。

数据驱动的思想是对于已知晓准确数据的值进行调整或覆盖。具体的实现技术有很多，包括定位错误信息及修正、数据标准/规范化、获取新数据等。当某个信息不准确、不完整时，可再次观察现实取得最新有效信息。当对这个信息的数据质量活动生效时，一些维度的质量会得到相应的改善。或者通过对象识别，对比原信息与高质量信息的差异。将待改进的信息与已知质量良好的其他信息进行比较。使用数据完整性约束也是提升数据质量的有效手段，通过该手段检测数据是否符合一致性，获取错误定位并采取纠正措施。

过程驱动侧重于过程，通过重新设计过程、生成或修改数据以提高其质量，主要关注的是衡量流程质量和制定流程改进提案。过程驱动的策略包括两个主要技术: 过程控制和过程重新设计。过程控制在从内部或外部源插入新信息、更新过程访问的信息源、涉及新信息源等情况下检查和控制过程插入信息生成过程，从而避免信息劣化和错误传播。过程重新设计策略是重新设计生产流程，消除影响因素的同时引入新活动，从而获取高质量信息。

2.5 系统开发技术栈

2.5.1 Redis

Redis 是一个在内存中的数据结构存储的开源高性能数据库，它作为跨平台的非关系型数据库，在许多实际应用中与关系型数据库起互补作用。在性能上，Redis 的读写速度分别高达每秒 11 万次和 8 万次。在数据结构上，它支持字符串、哈希、列表、集合和有序集合等多种数据类型。它支持 key-value 型数据存储，且提供原子性操作，即失败时维持执行前现状或者成功执行。为了获得最佳性能，Redis 使用内存中的数据集，并通过定期将数据集转储到磁盘或通过命令附加至基于磁盘的日志中来持久化数据。

2.5.2 Django

Django 为当前的主流的开源 Web 框架之一。Django 框架与 SpringBoot 框架类似，区别在于后者使用 Java 进行开发而 Django 使用 Python 开发。作为一个开源的框架，Django 最显著的优点就是有强大的社区和文档支持。

Django 具有强大的数据库功能，使用 Python 的类对应数据库中的表，即数据模型不依赖于特定的数据库，实现了数据库和数据模型的解耦。Django 的设计目的是可以简便快捷的开发 web 应用程序，重视代码重用，组件可以应用于整个系统，可迁移性强。同时，Django 还有一些实用的第三方插件，它的可扩展性也支持用户开发自己的作品。Django 可以使用 Redis 等缓存系统，网页的加载速度得到了保证，因此使用 Django 开发的 web 系统可以一定程度上保证用户体验。

2.5.3 Vue

Vue 是构建用户界面的渐进框架，它的设计结构为自底向上逐层应用。Vue 易于上手，且易于与其他第三方库或现有项目集成。另一方面，当与现代工具和支持库结合使用时，Vue 也完全能够支持复杂的单页应用程序。本项目使用 vue 框架实现前后端分离，通过 API 与后台交互。

2.6 本章小结

本章对项目所涉及的相关技术进行简单描述。首先介绍了与本文多维度量体系建立相关的数据质量领域的发展和研究现状，与事实模型构建相关的自然语言处理技术，包括了分词、词性标注、依存句法分析、文本摘要技术等。对于系统展示计算结果使用的文本数据可视化技术进行了简要介绍。接着介绍了可提升信息质量的信息质量活动及策略。最后，介绍了系统开发技术栈。

第三章　需求分析与概要设计

3.1　裁判文书可解释性分析系统整体概述

智慧法院建设是法院近年来的发展重点，利用人工智能技术在司法领域的应用辅助司法审判，如庭审语音识别技术代替手动输入文字材料，类案推送技术为法官筛选以往相似度较高的案例，案件结果预测为当事人提供分析信息。人工智能技术为司法流程中的每一环都节省了许多繁琐的程序及大量的时间，极大提高了司法工作人员的工作效率，为人民提供了便捷，可以快速获取想知道的信息。裁判文书作为每个庭审案件的精炼产品，自生成至出现于互联网的过程，涉及的人员众多，它所包含的内容、信息、质量时刻被关注着。

由于我国法院存在“案多人少”的现象，且民商事案件更是呈爆炸式增长的趋势，文书数量以指数级增长，法院负担繁重。裁判文书以文本的形式呈现，篇幅较长，阅读者需要耗费大量时间掌握内容获取信息。且面对数量逐渐庞大的文书，文书的受众更需要类似文书摘要的总结以快速找到目标文书。另一方面，高质量数据产生有效信息、创造更大的价值。然而数据质量问题随着数据量级增长逐渐显露，定位裁判文书的问题及提高其数据质量需要耗费高时间成本及人工成本。数据使用者需要借助工具提高使用效率。

为了解决上述问题，本文设计并实现了数据质量驱动的裁判文书可解释性分析技术，旨在针对裁判文书，提供高效可靠的文书及数据集可解释性分析和质量提升等功能。一方面，本技术希望通过提供智能化服务支持为文书数据相关工作者尽可能减轻工作量。另一方面，通过用户友好的操作界面，协助用户自主使用本技术自动解读单篇文书或文书集的核心内容信息，获得数据的可解释性分析报告，提高阅读效率，促进下一步研究。其中，细/粗粒度分析的事实模型的具体设计和多维度量体系的详细维度及指标的实现分别在度量解析模块中描述和解释。

3.2 裁判文书可解释性分析系统需求分析

3.2.1 涉众分析

表 3-1: 涉众分析

涉众类别	涉众特征与期望
普通阅读者	没有特殊能力要求且仅需通过文书了解相关案例。期望可以快速了解冗长文书的核心内容概要信息。
文书数据相关工作者	具有较强的行业领域知识和软件使用能力，负责裁判文书撰写或者检测文书质量，根据软件给出的可解释性分析报告，及时发现文书异常，对文书缺陷快速修复。期望利用工具减少人工工作量，加速工作进程，提供高质量裁判文书。
司法研究者	具有较强的行业领域知识，通常带着目的性有针对的广泛搜集数据，对所收集的数据运用数理方法进行分析并据此进行理论阐析。期望在数据时代下借助技术支持对数量充足的案例样本进行研究得出具有普遍性的结论。
政府工作人员	作为政府工作人员，希望可以通过大量的司法案例映射出社会问题，寻找解决之道。期望可以利用工具快速分析司法数据，多领域交叉融合，获得最大共赢。
数据测试人员	具有一定的专业测试能力和行业领域知识，熟悉各种计算机环境和常用软件测试工具，负责对依赖原数据集生成的新数据集进行详尽的测试，以判断数据是否符合客观规律及用户需求。期望更方便快捷地对扩增后的数据集进行测试，减轻工作强度，精准定位异常，进而提高测试效率。

本技术主要聚焦裁判文书可解释性分析，帮助裁判文书数据生产者、消费者快速了解单篇文书以及文书集的性质、概况等信息，因此涉众甚广，主要但不限于包括普通阅读者、文书数据相关工作者、司法研究者、政府工作人员、

数据测试人员。表3-1展示了各个类别的涉众特征与期望。随着裁判文书互联网化，广大人民群众都是裁判文书网的阅读者，普通阅读者可以通过本技术快速了解文书核心内容概要。文书数据相关工作者包括了法官、法官助手、裁判文书校对者、裁判文书审核者等人员，通过本技术提供的报告，针对性地发现文书缺陷，高效控制文书质量。司法研究者为法律实证研究者，如检察官、律师、高校老师，由于文书本身包含的信息并非严格意义上的数据或一些研究者所称的定量化的数据，本技术通过标签、编码等数据科学方法将案情信息转化为定量化数据协助实证研究，获将案件信息取精准数据报告从而衔接下一步工作。本技术的文书数据集可解释性分析报告为政府工作人员提供决策支持。数据测试人员需要了解数据增强后的数据集概况及是否符合某些内在规律。

3.2.2 功能性需求

本文从裁判文书本身的特性和使用者的角度亟需获取的特征信息考虑，为广大的可解释性分析服务的使用人员提供多维度的评估方案。数据质量驱动的裁判文书可解释性分析技术的主要功能性需求分为数据交互、度量解析、质量提升三部分。数据交互包括了数据上传、文件上传管理、问题文书管理、下载修复文书集、下载细粒度分析报告、下载粗粒度分析报告。度量解析分为启动细粒度分析任务、启动粗粒度分析任务、查看细粒度分析报告、查看粗粒度分析报告。质量提升的需求为修复问题文书。如表3-2所示。

3.2.3 非功能性需求

系统能否持续稳定并且高效的提供服务受非功能性需求影响。非功能性需求的缺失可能导致系统无法完全展示其优秀便捷的功能，用户体验差，从而埋没产品的功能性需求所带来的价值。为保障软件系统质量，系统还应满足下列非功能性需求。

(1) 可用性：系统全年正常运行时间应达到99%，即全年持续运行故障停运时间累计不能超过3天15小时36分。由于系统错误及其他原因引起系统崩溃时，能够恢复和还原数据。

(2) 易用性：本系统面向的人群对于软件技能要求不高，该需求涉及到人机体验，系统首页及各个功能模块的交互界面简洁明了，提供友好信息提示和指导，用户可快速知晓每一步操作从而达到目的。

表 3-2: 系统功能性需求列表

需求编号	需求名称	需求描述
R1	数据上传	用户可以通过浏览器客户端或系统接口上传需要解析的单篇文书或多篇文书，支持数据集形式上传。
R2	文件上传管理	用户拥有个人文件空间，可以对已上传的文书进行查看、删除等操作，提供搜索功能查找文书。
R3	启动细粒度分析任务	用户可在分析界面上传单篇文书或从已上传文件选择文书，选择待分析的维度指标后发送启动分析任务请求，等待结果反馈，支持用户在任务管理界面查看任务状态。
R4	启动粗粒度分析任务	用户可在分析界面上传文书集或从已上传文件选择文书集，选择待分析的维度指标后发送启动分析任务请求，等待结果反馈，支持用户在任务管理界面查看任务状态。
R5	任务管理	用户在任务管理界面查看任务执行状态，并且可随时查看执行完毕的任务的任务报告。
R6	查看细粒度分析报告	支持对用户自定义选择的度量维度计算结果自动化布局并在线展示。
R7	查看粗粒度分析报告	支持对用户自定义选择的度量维度计算结果自动化布局并在线展示。
R8	下载细粒度分析报告	支持用户下载分析报告至本地查看。
R9	下载粗粒度分析报告	支持用户下载分析报告至本地查看。
R10	问题文书管理	支持用户对分析任务过程中检测出的问题文书进行管理，确认是否符合问题文书标准。可查看、删除或修复。
R11	问题文书修复	帮助用户对问题文书列表的文件自动化修复，并更新到文件管理中。
R12	下载修复数据集	用户在完成分析任务且修复问题文件后，可选择更新的数据集。

(3) 鲁棒性：数据格式及内容的多样性易使系统出现异常或停止，系统应该具有自动恢复正常能力且将错误原因反馈给用户。

(4) 及时性：面对多用户并发请求可降低延迟，快速响应，页面响应和加载时间应该控制在 1 秒以内。

(5) 安全性：确保数据安全，为用户分配合适的权限并保护用户数据，无法访问其他用户的数据，防止泄露。系统保证只有管理员可以更改权限。

(6) 可扩展性：系统功能模块化，面对负载增长或系统迭代及修改，可快速完成；度量方法的各指标应降低耦合度，支持用户自行增加其他度量方法。

3.2.4 系统用例分析

表 3-3: 系统用例列表

用例编号	用例名称	功能需求编号
UC1	数据上传交互	R1
UC2	文件管理	R2、R12
UC3	启动分析任务	R3、R4
UC4	分析报告交互	R5、R6、R7、R8、R9
UC5	问题文书管理	R10
UC6	问题文书修复	R11

涉众分析展示了五类本系统的潜在使用人员，普通阅读者更偏向于单篇文书的解读，文书数据相关工作者基于单篇文书以提高文书质量，某些特定场景下也需审核文书数据集质量，司法研究者与政府工作人员更关心整个文书数据集所呈现出的现象，数据测试人员则需要从粗粒度角度判断增量数据集的大致情况，进一步从细粒度角度寻找异常文书。

尽管上述五类人群的使用功能侧重点不同，但用户在系统的功能使用中并无区别。根据上述涉众分析、功能性需求分析、非功能性需求分析，我们总结类如图3-1所示的系统用例图，涉及 6 个用例，分别是数据上传交互、启动分析任务、分析报告交互、问题文书管理、问题文书修复、下载修复数据集。

本小节通过介绍参与者、三种条件以及两种流程详细阐述了每个用例。涉及用例和功能需求的对应关系如表3-3所示，其中文件管理对应了文件上传管理和下载修复数据集两个需求，启动分析任务包含了启动细粒度分析任务和启

动粗粒度分析任务，分析报告交互分为查看细粒度分析报告、查看粗粒度分析报告、下载细粒度分析报告、下载粗粒度分析报告。

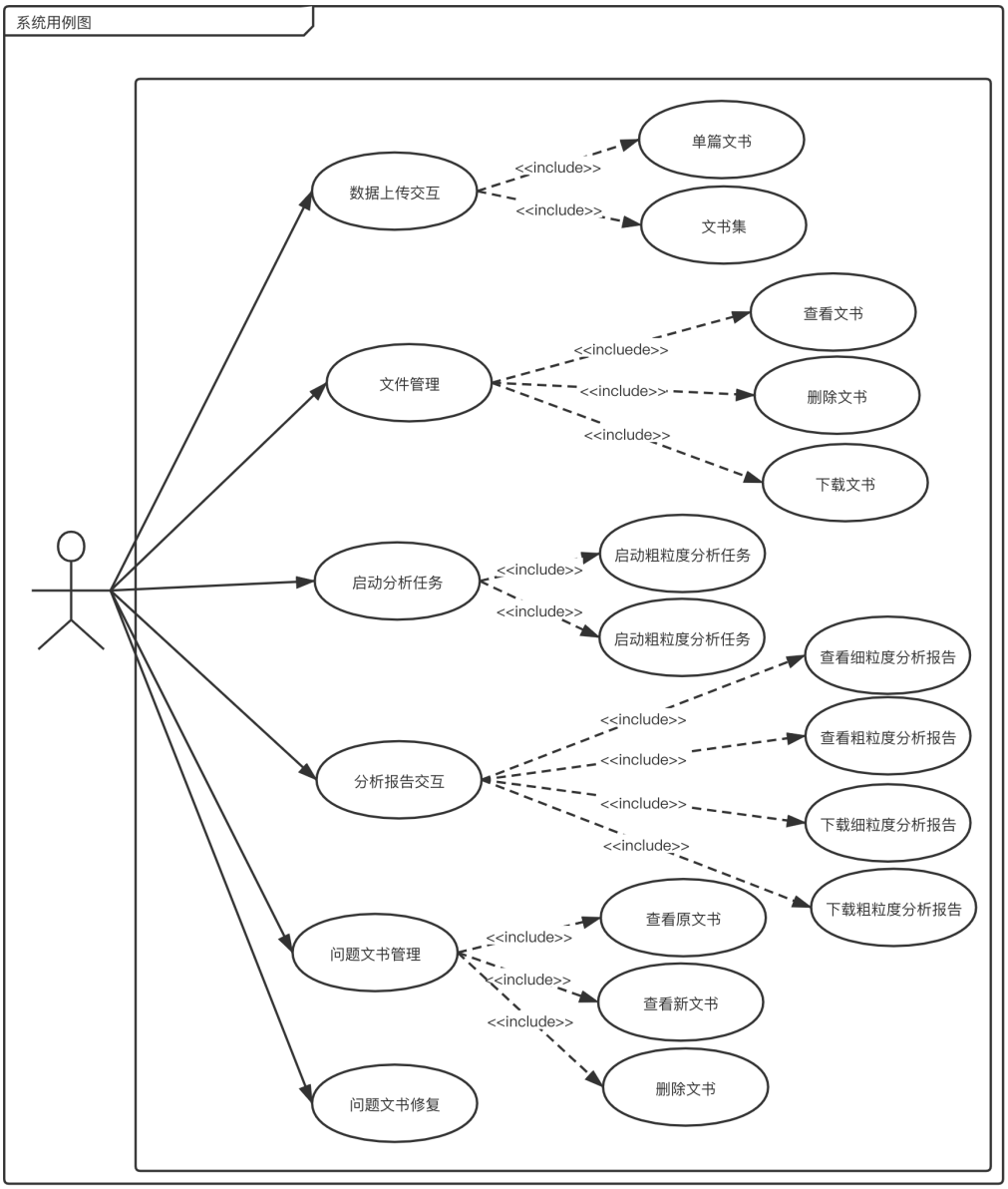

图 3-1: 系统用例图

表3-4为数据上传交互用例，用户登陆系统后，可主动点击文件上传或选择分析任务时触发，进入数据上传界面，选择待测单篇/多篇文书或文书数据集。发送上传请求后，系统对待上传数据自动化审核，出现数据格式异常或其他异常等情况则中止请求，确认上传后后台将文件上传至文件系统并对本次上

传的文件初步处理，将持久化信息存入数据库以待进一步操作。上传完毕后用户可到文件管理界面查看已上传的数据。

表 3-4: 数据上传交互用例描述

ID	UC1
名称	数据上传交互
参与者	用户
触发条件	用户处于数据上传界面，点击相应按钮发送请求
前置条件	1. 用户身份完成识别并拥有权限 2. 上传数据格式符合系统要求
后置条件	成功上传的数据存储于文件系统并反馈信息
正常流程	1. 用户处于数据上传界面，选择数据所在路径 2. 发送上传请求 3. 系统判断是否符合标准，符合则存至文件系统 并将文件预处理信息持久化至数据库 4. 返回成功上传信息，包括文件名、文件大小等

表 3-5: 文件管理用例描述

ID	UC2
名称	文件管理
参与者	用户
触发条件	用户完成数据上传后跳转或点击个人文件管理进入文件管理页面
前置条件	用户身份完成识别并拥有权限，可以对用户在系统中所拥有的文件进行操作，如查看、删除或下载
后置条件	根据不同操作返回相应信息
正常流程	1. 用户处于文件管理界面，系统显示文件列表 2. 用户点击文书对应的查看按钮，系统返回文书详细内容 3. 用户点击文书对应的删除按钮，系统删除该文书并返回“删除成功”提示，重新跳转回文件管理界面 4. 用户可以单选/多选/全选下载单篇文书/多篇文书/全部文书，系统将所选文书打包下载 5. 用户在搜索框输入文书名称，系统返回该文书

表 3-6: 启动分析任务用例描述

ID	UC3
名称	启动分析任务
参与者	用户
触发条件	用户在文件管理页面中根据需要选择细/粗粒度，点击对应的分析按钮
前置条件	1. 用户身份完成识别并拥有权限 2. 用户文件系统中至少有一篇文书
后置条件	将生成报告存入数据库并在前端显示且支持下载
正常流程	1. 用户处于文件管理界面，选择细粒度解析则点击文书相应分析按钮；选择粗粒度解析则选择相应文书后点击粗粒度解析按钮 2. 进入细/粗粒度分析界面，选择所需指标 3. 点击确认按钮启动分析任务，系统对单篇文书完成预处理并调用指标接口，生成细粒度分析报告；系统对文书集的所有文书完成预处理并调用指标接口 4. 系统生成细/粗粒度分析报告

表 3-7: 分析报告交互用例描述

ID	UC4
名称	分析报告交互
参与者	用户
触发条件	在分析任务成功执行后的反馈界面点击查看分析报告，可在线浏览或下载报告至本地
前置条件	1. 用户身份完成识别并拥有权限 2. 用户启动分析任务并成功生成分析报告
后置条件	前端返回报告详情且提供报告下载功能
正常流程	1. 用户处于任务管理界面，点击刷新按钮查看任务是否完成 2. 系统分析任务完成后改变任务状态，并返回查看报告按钮 3. 用户点击查看报告按钮，系统显示分析报告详情 4. 用户点击报告中的下载按钮可以在本地查看分析报告

文件管理用例如表3-5所示，系统支持用户对已上传的文件进行管理，包括查看、删除、下载功能。界面内提供搜索框查找文书，用户输入文书名称，返回该篇文书。用户进行质量提升服务后，用户可以在文件管理处查看更新的文书。系统提供数据下载功能，用户可以对现在的文件进行单选或多选，系统自动打包文件并下载至用户本地。

表3-6描述了启动分析任务用例。用户确定文书解析粒度后进入特定界面，上传待分析数据或选择已上传数据作为本次任务的主体。待分析数据确认完毕后，根据自定义需求选择度量解析的指标组合，启动分析任务。分析任务启动后，自动触发文本预处理流程，按照规则完成结构解析。事实模型和多维度量体系为本技术的核心，每个维度下细分多个指标。系统调用各维度度量服务计算结果，存储于 redis 中，由分析报告展示服务对各部分结果聚合调用，自动化布局后在线展示。

本用例（见表3-7）描述用户在分析任务成功执行后获取结果的步骤。结果以分析报告形式展示，用户与之进行交互。用户可在任务管理界面的相应分析任务处点击查看报告在线浏览或下载报告至本地浏览。

表 3-8: 问题文书管理用例描述

ID	UC5
名称	问题文书管理
参与者	用户
触发条件	用户进入问题文书管理界面
前置条件	1. 用户身份完成识别并拥有权限 2. 用户成功执行分析任务
后置条件	根据用户操作反回相应内容或提示信息
正常流程	1. 用户进入问题文书管理界面，系统显示问题文书列表 2. 点击查看文书的相应问题详情列表，查看后返回 3. 点击相应文书的原始文书查看按钮，确认其是否需要修复 4. 用户确认文书无异常，点击相应文书的删除按钮 5. 点击相应文书的新文书查看按钮，查看修复后的文书详情

表3-8为问题文书管理用例描述。在分析任务执行过程中，系统同时触发检测问题文书服务，任务结束后，将可疑的文书归入问题文书管理，待用户进

一步确认是否需要修复。无需修复的文书可由用户删除移出问题文件列表。所有问题文书均提供对应的问题详情列表，包括存在问题的修复建议。待修复文件提供三种选择操作：查看，修复，删除。已修复文件提供三种选择：原文件查看，新文件查看，删除。值得注意的是，对于部分问题本技术无法直接修复，有待人工校正。因此，对于仅存在本类问题（即第二类质量问题）的文书的无法使用修复机制。

本技术的最终目的是帮助用户了解数据以及数据集情况，协助用户侦查异常或不符合需求的数据，尽可能提升数据质量，从而充分发挥数据价值。表3-9为问题文书修复用例描述。问题文书修复功能的目的是为了减少用户在数据处理方便的工作量，尽可能处理已检测的问题。

表 3-9: 问题文书修复用例描述

ID	UC6
名称	问题文书修复
参与者	用户
触发条件	用户处于问题文书管理界面并点击相应文书的修复按钮
前置条件	1. 用户身份完成识别并拥有权限 2. 用户成功执行分析任务
后置条件	系统修复问题文书并生成新文书存储至文件系统
正常流程	1. 用户进入问题文书管理界面，系统显示问题文书列表 2. 点击相应文书的修复按钮，启动文书修复任务 3. 系统修复文书并存储至文书系统，返回修复成功提示信息
扩展流程	3a. 启动失败 1. 系统自动重新发起请求，再次失败则返回错误记录 2. 向用户反馈失败提示信息

3.3 裁判文书可解释性分析系统概要设计

3.3.1 系统总体架构设计

本章前序部分完成了涉众分析，功能性需求分析，非功能性需求分析以及用例分析，最终得到的系统整体框架图如图 3.3 所示。本系统采用了前后端分

离的设计方式，用户通过浏览器访问裁判文书分析平台，根据用户操作调用文件管理、文书分析等服务。本部分将对框架内交互的各模块进行简要介绍。

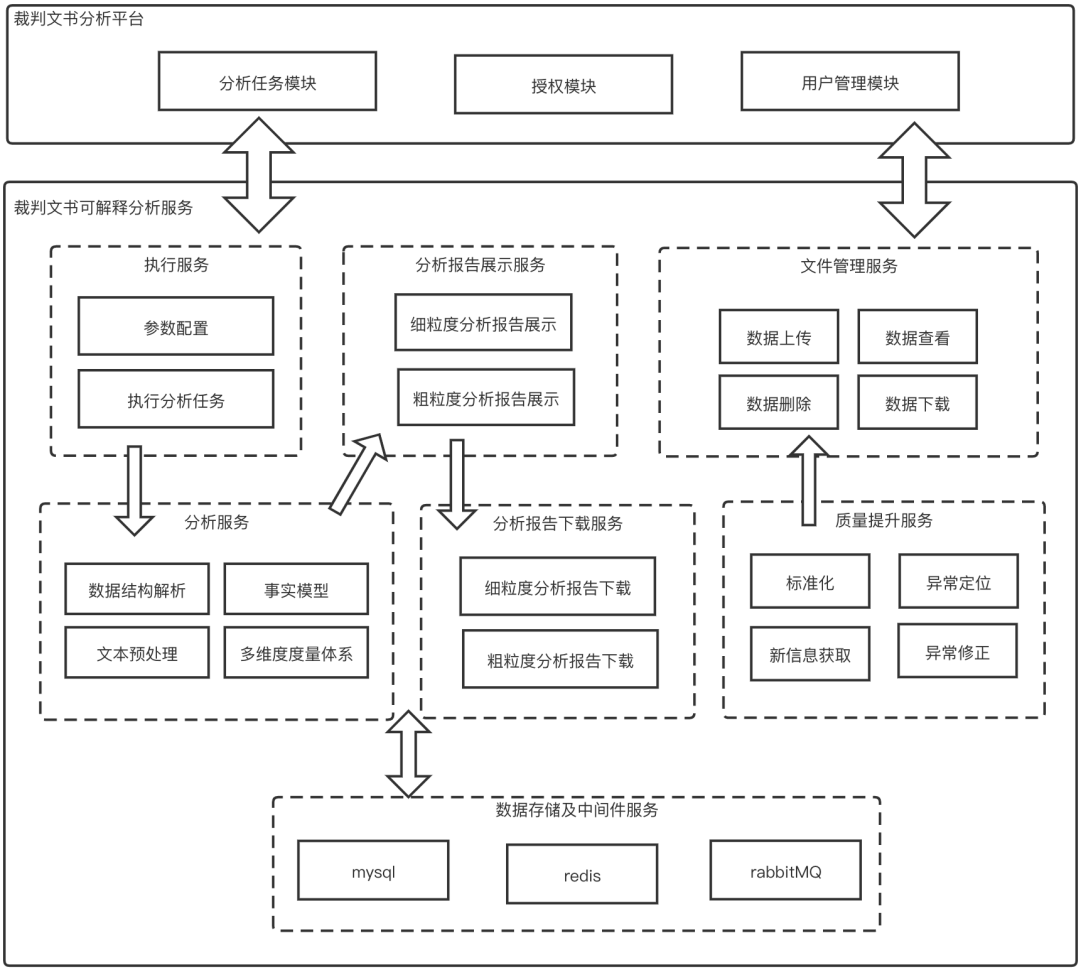

图 3-2: 系统整体架构图

裁判文书分析平台包括了用户管理模块、授权模块以及核心的分析任务模块。用户管理模块调用文件管理服务，用户在此处完成数据上传，并且对已上传的数据具有查看、删除、下载的权限。用户权限由设置好的授权模块自动化配置，只有管理员可以访问并修改授权模块从而保障系统的安全。用户登陆获得授权后可直接访问分析任务模块，该模块调用执行服务，根据用户选择待分析的文书或文书集，确定所需的分析维度，系统调用分析服务，首先对文书完成数据结构解析和文本预处理，使用设定的事实模型和多维度量体系完成分析服务。事实模型和多维度量体系具有可扩展性，用户可根据需要开发新维度并聚合。分析服务完成后，调用分析报告展示服务，聚合计算结果并自动调整布

局反馈至前端。用户可选择分析报告下载服务在本地查看报告。质量提升服务在分析服务结束后调用，系统在使用分析服务过程中自动检测出问题文书，在问题文书管理处生成列表待用户进一步确认。用户可调用质量提升服务对问题文书进行修复，修复后的文书更新至文件管理处。完成分析任务和质量提升后，用户可以下载更新的数据。

3.3.2 架构建模

一个复杂的系统只有从多个不同视角进行设计解读才能形成合理的抽象描述，于是 Philippe Kruchten 于 1995 年提出了“4+1 视图”[53]，引起了业界的极大关注，并成为了现在软件设计的结构标准。“4+1 视图”采用了“分而治之”的办法，分别由逻辑视图、开发视图、进程视图、物理视图四类视图和场景视图组成，系统组织成员围绕这几个视图进行设计、实现、验证，并不断迭代修改。场景视图是从用户的角度确认需求及业务场景，上文已对此展开了详细介绍，此处不再赘述。下文将围绕四类视图对系统整体架构展开介绍。

1. 逻辑视图

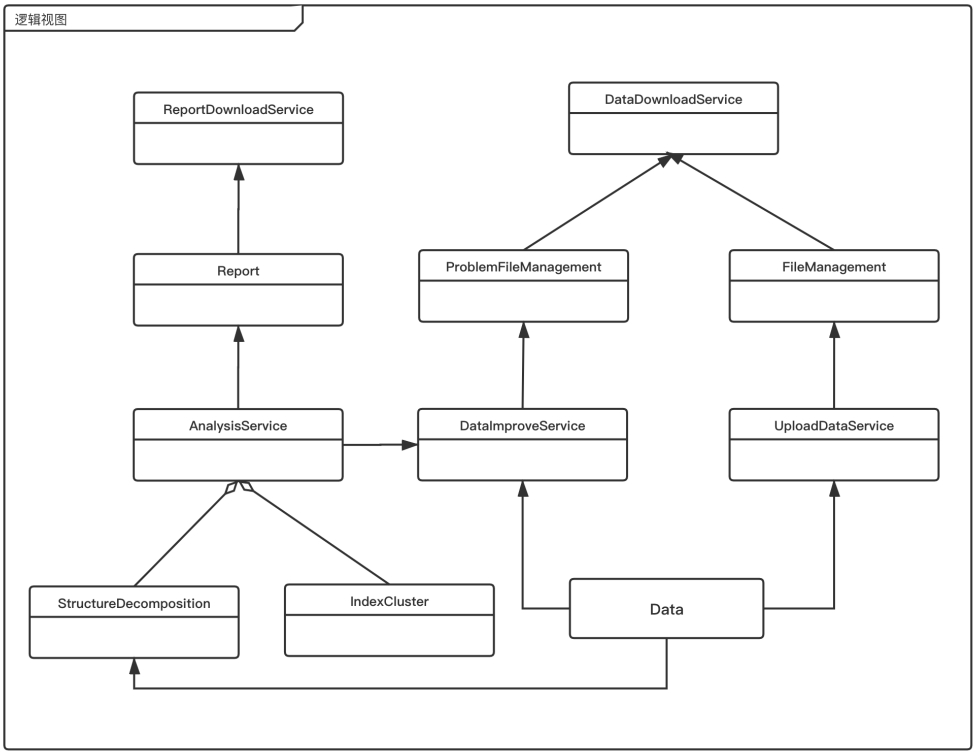

图 3-3: 逻辑视图

逻辑视图展示的是系统应向用户提供的服务，即功能需求。使用面向对象的思想，利用抽象、继承和封装的原理分解，分解对象或对象类的关系。通常使用类图来表示逻辑结构，而受数据驱动的应用程序也可以使用如 E-R 图来替代。我们抽象分解了系统的功能需求，逻辑视图如3-3所示。*Data* 为本系统的主要操作对象即裁判文书，*UploadDataService* 和 *DataImproveService* 均与 *Data* 直接关联，前者负责数据上传服务，上传后的数据可通过 *FileManagement* 完成文书管理；后者对部分问题数据进行修复，问题文书通过 *ProblemFileManagement* 管理。*DataDownloadService* 提供文书数据下载服务，用户可根据随时需要下载已上传数据，或者下载修复后的文书。系统利用 *StructureDecomposition* 在数据上传的同时处理数据信息并持久化存储。*AnalysisService* 由 *StructureDecomposition* 和 *IndexCluster* 组成，启动任务分析时需要文书数据的相关信息及根据用户选择的配置调用指标集合，将计算结果传送至 *Report* 生成分析报告，*ReportDownload-Service* 在用户需要时负责报告下载服务。

2. 开发视图

开发视图是以开发编程人员视角，设计系统实际软件模块组织，采用“分而治之”的思想将系统拆分为多个程序块分开实现，方便人员分配。子系统通常使用分层结构组织，每一层均为上层预留良好定义的接口。开发视图既包含了源程序，也包含可调用的第三方 SDK 和现成框架、类库，以及将运行于开发系统之上的中间件平台。

如图3-4所示为本系统的开发视图，UI 部分由页面、组件、以及其他静态资源构成。*Server* 部分采用分层组织架构，自上而下分别为 *Controller* 层、*Service* 层、*Respository* 层。*Controller* 层负责向外部提供服务，上传文件接口由 *UploadFileController* 提供，分析任务管理接口由 *TaskController* 提供，文书管理接口由 *FileManageController* 提供，问题文书管理接口由 *ProFileManageController* 提供，分析任务执行接口由 *AnalysisController* 提供，分析报告查看及下载接口由 *ReportController* 提供，问题文书修复接口由 *ImproveController* 提供。*Service* 负责完成 *Controller* 接收处理后的外部请求，执行实际的服务。*Repository* 层包括 *RequestRepository*、*TaskRepository*、*FileRepository*、*ReportRepository*，提供了对接底层数据库的操作支持，完成数据的存储和查询等任务。

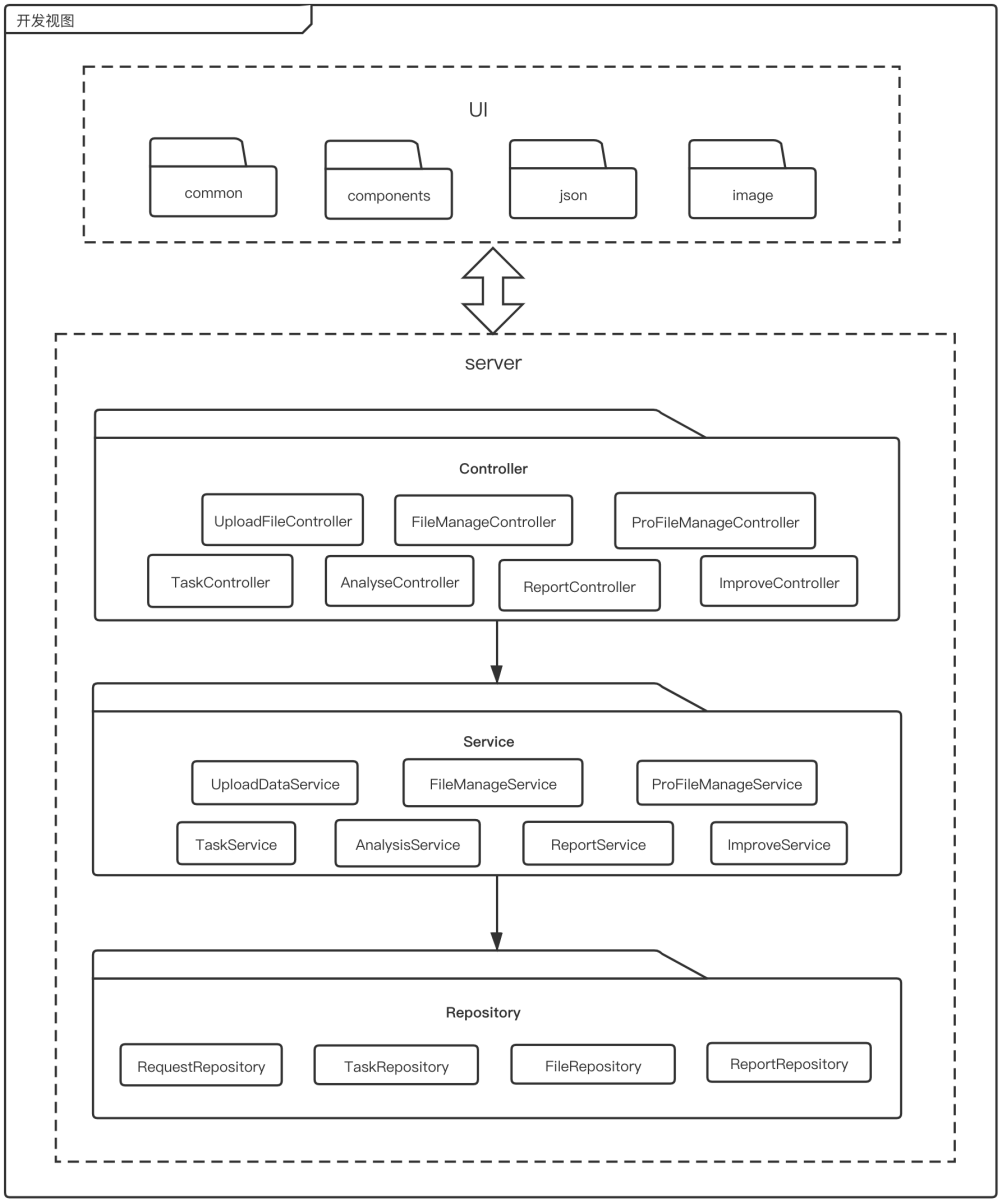

图 3-4: 开发视图

3. 过程视图

过程视图主要任务是捕捉设计的并发和同步特征。关注系统的非功能性需求如性能和可用性，考虑并发性、分布性、容错性等问题。过程是形成可执行单元的一组任务，过程表示为控制过程结构体系结构级别的策略，包含了过程的启动、恢复、重新配置和终止。过程的复制可达到增加处理负载的分配或提高可用性的效果。如图3-5所示为分析服务的过程视图。执行服务进程使用RabbitMQ 消息队列发送操作消息重复调用分析服务主进程以分析消息队列中

的每篇文书。分析服务主进程进而通过操作消息从 Redis、Mysql 以及文件系统获取分析过程中采集的中间数据和源数据，调用结构解析及多维度量体系的具体方法以对待测文书进行完整分析，解析及计算结果持久化存储，可供后续查询使用。待测文书集分解为单篇文书分析子任务，待子任务执行完毕后启动分析服务主进程对文书分析后的信息聚合，结果存入数据库，待报告生成服务调用反馈给用户。

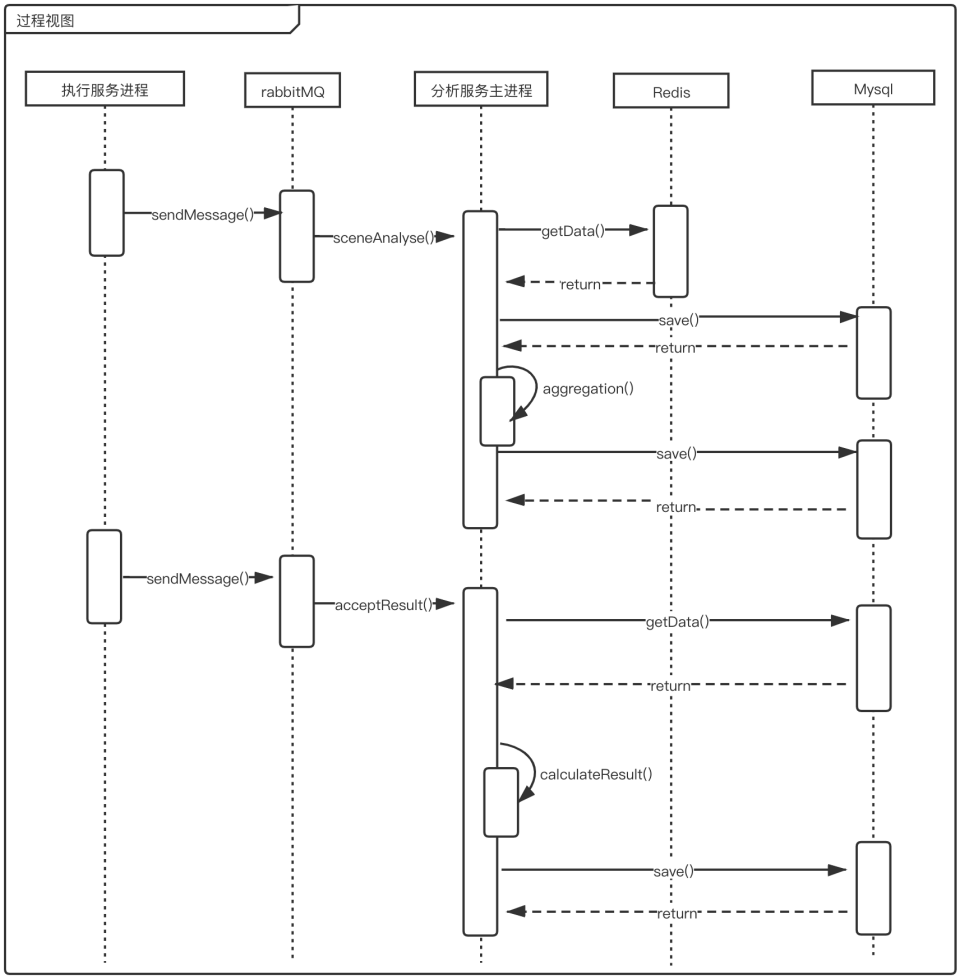

图 3-5: 过程视图

4. 物理视图

物理视图展示的为软件到硬件的映射，又称为部署视图。设计的软件系统依赖于配置环境，最终运行于物理或软件环境上。物理视图是从物理层面考虑系统拓扑结构以及物理环境如服务器、移动终端等物理设备，反映系统在分布式角度的设计。用户通过浏览器发送 http 请求访问系统，执行服务后端服务器

根据用户指令向消息队列服务器发送操作消息，消息队列服务器均通过 TCP 连接执行服务后端服务器以及分析服务后端服务器，分析服务后端服务器接收并执行请求时通过 HTTP 与数据服务器建立连接获取分析任务的前置数据。

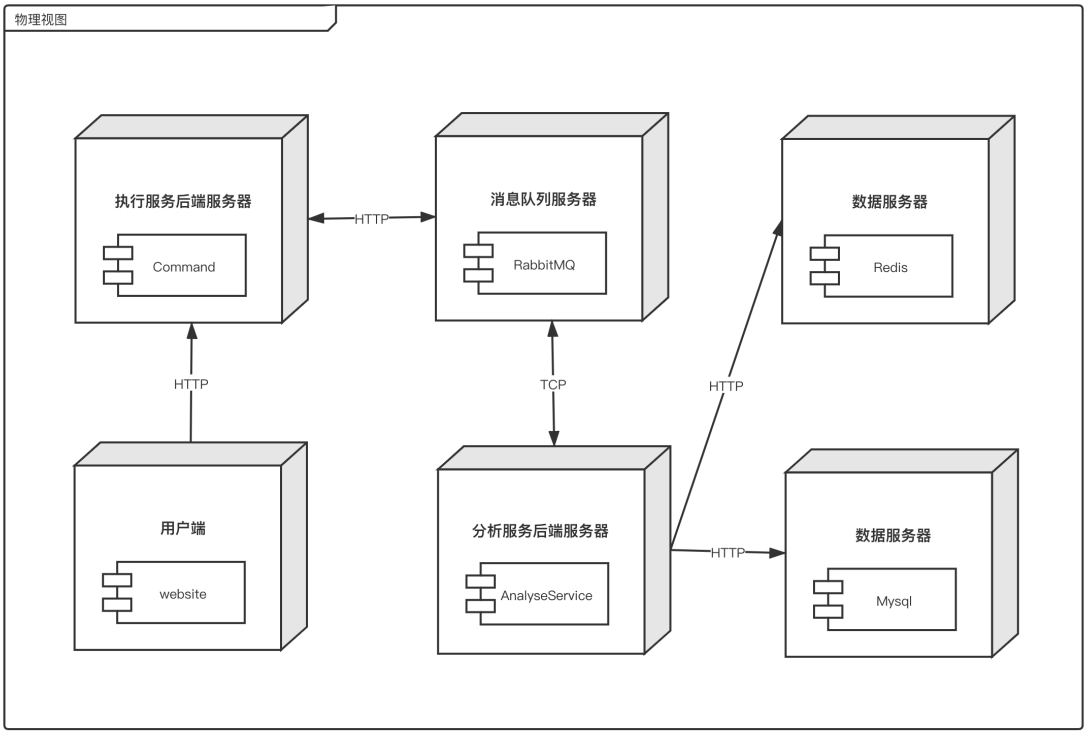

图 3-6: 物理视图

3.4 数据交互模块

该模块为用户与系统交互模块，包含用户上传文件、查看、删除、下载文书集等功能，核心类图如图 3-7所示。用户启动 *upload_service* 请求数据上传服务时，触发用户身份校验和权限校验。*upload_request* 负责用户数据上传，依赖于 *upload_service*，调用 *upload_deal* 对用户待上传的文件进行解析、校验，获取用户信息，将处理后的文件存储至用户的文件空间，文件解析获取信息持久化至及用户拥有的数据库空间下。*file_pre* 为用户上传文件的映射信息，由于文书的部分信息后续会直接或间接使用，因此 *file_pre* 包含了文书的事实模型部分信息，并通过文件名关联文件。*manage_service* 负责用户文件管理，用户通过映射信息关联文件对文件进行管理，系统支持删除已上传的文件，支持用户下载其用户空间下的文件。

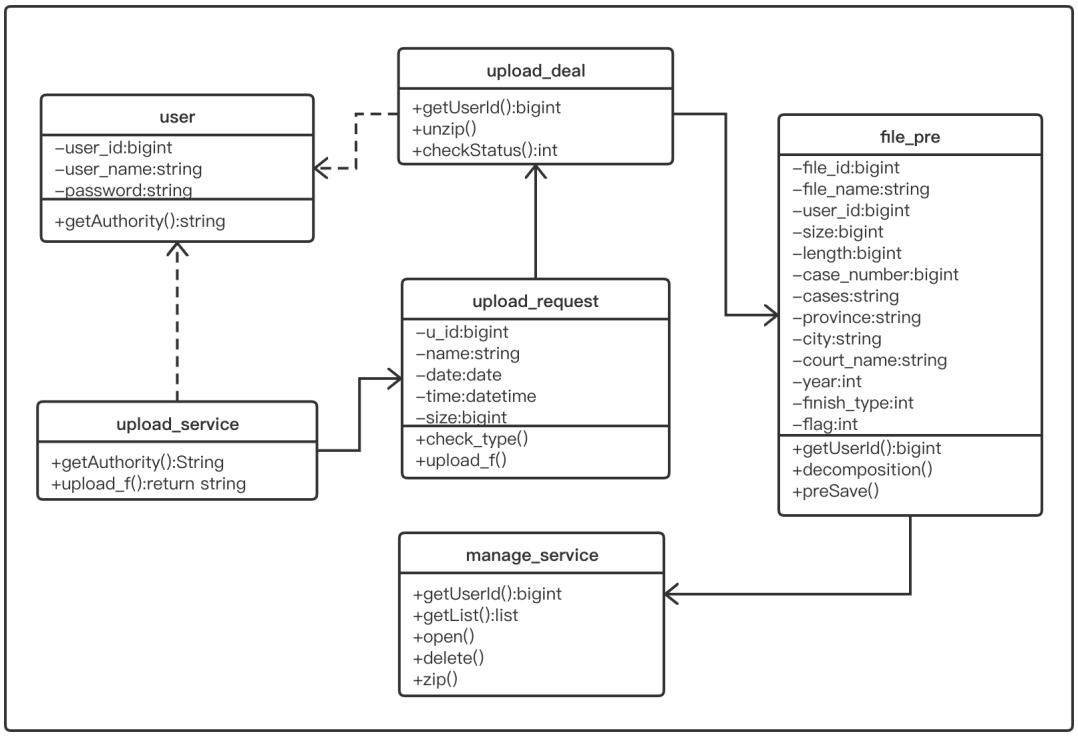

图 3-7: 数据交互模块设计类图

3.5 度量解析模块

度量解析为本技术的核心，本部分分别从文书粒度和文书集粒度层面建立多维度量指标体系及质量提升体系，主要从事实和数据/信息质量多个维度分析文书和文书集。下文为了方便，将“文书粒度”记为细粒度，“文书集粒度”记为粗粒度。细粒度和粗粒度解析均基于文书的结构解析后的标签粒度。

度量解析模块的核心类图如图3-8所示，*create_task* 负责用户创建分析任务，与此同时需确定数据来源，通过 *file_pre* 关联信息选择已上传文件作为本次任务的数据来源或者直接调用 *upload_service* 直接上传本次任务待分析文件。*task_request* 为实际运行任务的类，接受任务配置参数调用 *index* 的指标方法进行结果计算，创建 *report* 对象，将结果存入具体对象中持久化存储至数据库。*task_request* 接收 *report_id* 并更新任务状态。用户可在任务管理界下载已完成任务的分析报告，也可以删除等待中或已完成的任务。

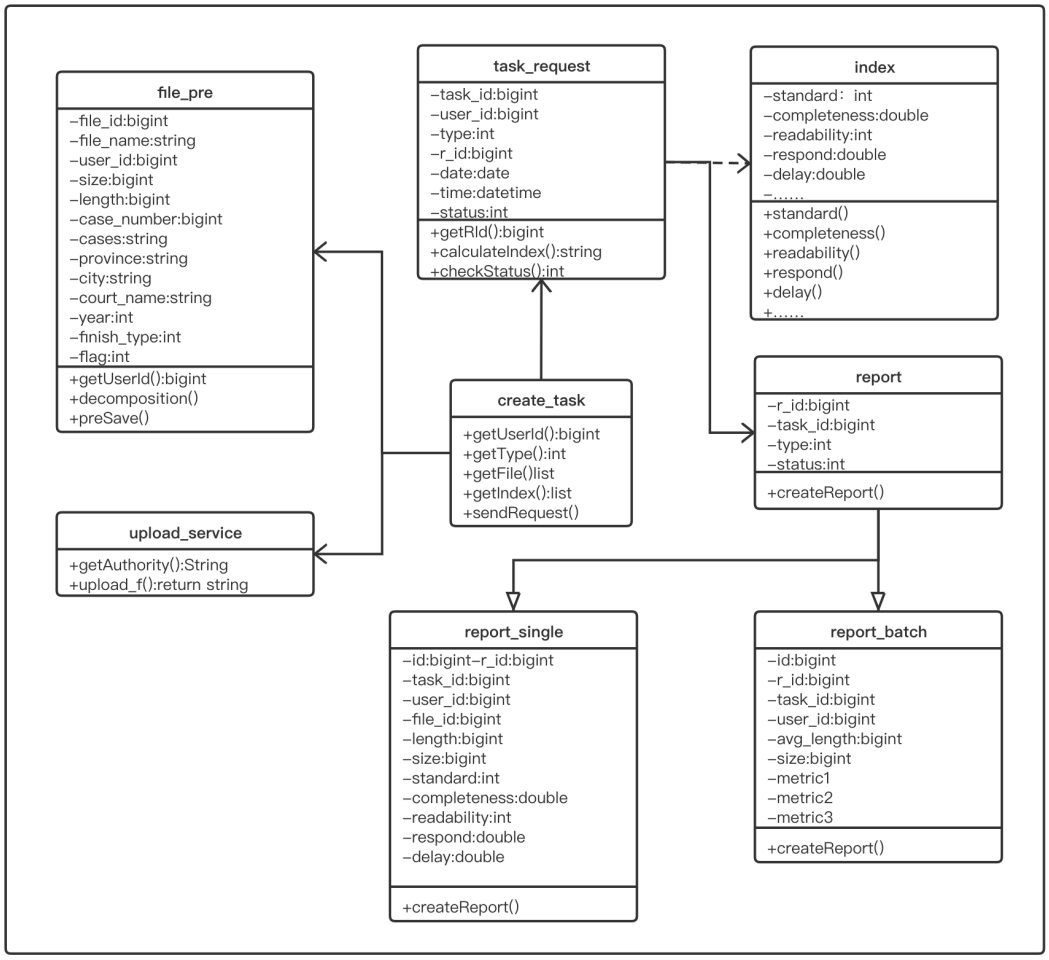

图 3-8: 度量解析模块设计类图

3.5.1 文书结构解析

根据最高院发布的《人民法院民事裁判文书制作规范》、《民事诉讼文书样式》(下述分别称为《规范》和《样式》) 及现有的民事裁判文书，裁判文书结构普遍固定，总结文书主要构成部分如表3-10。根据多年的文书统计，民事裁判文书均占比每年所有文书数量的一半以上，因此本文主要面向民事裁判文书。以民事一审裁判文书为例，主要由文首、当事人、诉讼记录、案件基本情况、裁判分析过程、判决结果、文尾七个主要部分组成。

文首具体为经办法院、案号以及文书的性质等详细信息；当事人部分包含了起诉方、代理人、应诉方的详细信息，可能为个人或单位；诉讼记录包含了案件审理的过程信息；案件基本情况为原告和被告分别陈述，法院根据双方陈

述及各自提供支持陈述的证据查明事实；裁判分析过程是整个案件涉及的法律法条；判决结果包含了结案方式和判决结果内容，诉讼费用承担，是否支持原告诉讼请求等详细信息；文尾表明案件的裁判时间以及审判组织成员以表公正。部分文书还包含附件。

3.5.2 细粒度分析

尽管可以总结出单篇文书的大致格式，但由于文书来自各级人民法院，质量不一，文书格式规范性方面仍有待提高。相比于 1992 年, 最高院在 2016 年修订的《样式》及《规范》中，文书样式种类在数量上由 372 种增加至 568 种。

据统计，文书篇幅普遍较长，甚至存在长达 12 万字的文书。且裁判文书的篇幅在近几年有越来越长的趋势。随着裁判文书上网，文书数量指数级增长。面对数以万计的可选择阅读的裁判文书数据，阅读者显然希望以高效率获取文本有效信息，最快作出最优选择。

根据 [54] 所述裁判文书的具体结构和文风尽管因案而异，仍应包含以下五大要素，本部分呈现文书内容特征。

（1）关于基本案情、程序流程和裁判结果的介绍性陈述;

（2）关于判决解决的争议焦点的陈述;

（3）关于案情的基本描述;

（4）关于如何适用相关法律解决案件争点的论证;

（5）案件判决结果。

为尽可能还原文书真实度，从事实角度展示了几大内容特征，包括了文书性质、案件特征、案件情况描述、时间线。本文结合国标及司法特性，从数据/信息质量角度选择了规范性、完备性、可读性、响应性、延迟性、平等性多个维度评估裁判文书。

事实

根据《制作规范》和《样式》的要求，本部分尽量以精简的文字和结构体现文书精要。事实部分包括文书文件的性质信息以及内容的特征，文件性质包括文件名，文件大小，文书长度。内容特征包括法院名称、文书名称、案号、地区、年份、案件由来、立案日期、案件基本情况、裁判分析过程、判决结果、时间线等。以上信息为分析裁判文书的基础。

表 3-10: 民事裁判文书结构

模块名称	模块内容	说明
文首	经办法院	包含法院层级代码、标准法院名称、法院级别、行政区划（省）、行政区划（市）
	案号	包含了字号、立案年度、法院简称、案号顺序号
	其他	包含文书制作单位、法院文书种类、文书名称、案件类别、文书种类、审判程序、案件类型
当事人	起诉方	包含了诉讼参与人、诉讼身份、单位等信息
	代理人	包含了诉讼参与人、诉讼身份、单位等信息
	应诉方	包含了诉讼参与人、诉讼身份、单位等信息
诉讼记录	开庭审理信息	是否开庭审理、是否公开审理等信息
	开庭日期	格式为 xxxx 年 xx 月 xx 日
	案由	包含完整案由、案由代码等信息
	受理日期	格式为 xxxx 年 xx 月 xx 日
	立案日期	格式为 xxxx 年 xx 月 xx 日
	其他	包含一审案件适用程序、独任审判、简易转普通、一审案件来源、审判组织等信息
案件基本情况	原告诉称段	原告陈述事实及提出诉求
	证据段	原告提供相关证据，包括证据的名称、种类、提交人
	被告辩诉段	被告陈述事实为自己辩诉
	证据段	被告提供证据反驳原告
	查明事实段	法院审理查明事实
裁判分析过程	法律法条引用	拆分为名称、条、款
	法律法条分组冗余	格式为《xx 法》第 xx 条（第 x 款）
判决结果	一审结案方式	包括判决、撤诉等结案方式
	判决结果内容	包括判决责任承担方式、权利人、义务人等信息
	诉讼费承担	包含诉讼费承担金额、承担人、交纳情况及结案标的额
	其他	包括提出管辖权异议、是否支持原告诉讼请求等信息
文尾	裁判时间	结案时间
	审判组织成员	包含审判人员姓名和角色

文书内容包含了案件经过及裁判事项，属于长文本，直接展示过于冗长，因此本文使用文本摘要技术展示本部分内容。文本摘要技术根据摘要目的的不同，分为关键词摘要、短语摘要、句子摘要、段落摘要。关键词抽取是目前非结构化文本创建摘要时最常使用的技术，通过呈现关键字、关键字频率和关联性以达到有效视觉表示。在大文本数据分析中，词云可以快速感知最突出的文字，但是显然它并不适用于裁判文书的场景下。裁判文书篇幅有限，关键词频率区分度不大，且司法数据使用者一般要求精炼且准确。散乱的无关联的词汇展示还需浏览者自行想象关联，容易出现歧义。针对此类非结构化文本创建摘要，可利用段落摘要，结合语义对关键部分实现缩减提炼，使用抽取与生成的方式获取段文本，从而达到快速高效掌握文本数据信息。

对文书的时间线梳理在一定程度上帮助用户快速了解案件的来龙去脉，本文通过时间节点及其事件展示案件时间线。运用历史方法，以时间为主轴整理证据信息，形成点状的证据事实，以大事记形式展现出来，清晰完整呈现案件相关时间节点，并结合词性标注和命名实体识别技术剥取事件主干信息。

规范性

在国标中，规范性的定义是数据符合数据标准、数据模型、业务规则或权威参考数据的程度。如表3-11所示，本维度包含了标点符号规范性、数字规范性、案由规范性、来源规范性、适用程序规范性、裁判依据规范性等多项指标。规范性要求文书结构清晰，语言格式恰当。对案件、证据、理由和最终结果的清晰阐能提高判决书的清晰度。裁判文书不得有书写或打字错误，语法也必须正确。法官正确引用法律是对法官基本的要求。如裁判依据中容易出现引用法律条文有误，直接引用宪法，甚至引用地方文件或某些会议既要等问题。

完备性

完备性是指数据是充分的，任何有关操作的数据都没有被遗漏。完备性具有丰富的内涵，包括数据集的完备性、记录的完备性、字段的完备性、字段内容的完备性等。本文针对裁判文书的完备性定义是裁判文书的主要结构是否完整，必要项是否存在，由完整度体现文书的完备性。完备性的指标提出及计算参考文书结构——民事裁判文书结构表3-10, 通过统计该结构表的模块完备度计算文书完备度。

表 3-11: 规范性指标

指标类别	具体内容
标点符号规范性	1、“被告辩称”“本院认为”等词语之后用逗号； 2、“××× 向本院提出诉讼请求”“本院认定如下” “判决如下”“裁定如下”等词语之后用冒号; 3、裁判项序号后用顿号; 4、其他标点符号用法按照标准 GB/T15834-2011 执行。
数字规范性	1、案号使用阿拉伯数字； 2、裁判尾部落款时间使用汉字数字； 3、裁判主文即列举阐述的序号使用汉字数字； 4、其他数字用法按照标准 GB/T15835-2011 执行。
法院名称规范性	基层、中级人民法院名称前应冠以省、 自治区、直辖市的名称，但军事 法院等专门人民法院除外。
案号规范性	案号由收案年度、法院代字、类型代字、 案件编号组成。
案由规范性	应当准确反映案件所涉及的民事法律关系的性 质，符合最高人民法院有关民事案件案由的规定。
来源规范性	来源包括八项：新收；有新的事实、证据重新起诉; 上级人民法院发回重审; 上级人民法院指令立案受理; 上级人民法院指定审理; 上级人民法院指定管辖; 其他人民法院移送管辖; 提级管辖。
适用程序规范性	包括普通程序、简易程序、小额诉讼程序和非讼程序。 非讼程序包括特别程序、督促程序、公示催告程序等。
裁判依据规范性	引用最高人民法院的司法解释时，按照公告公布 的格式书写；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；引用法律、法规、司法解释应 书写全称并加书名号；引用法律、法规和司法解释 条文有序号的，书写序号应与法律、法规和司法 解释正式文本中的写法一致。

$$Completeness(X) = \frac{1}{n}\sum \frac{P_i(s|X)}{P_i(s|U)} \tag{3-1}$$

U 代表裁判文书模块全量信息集，X 为该文书模块信息集，s 为在该信息集内基本结构项，则 P(s|U) 为全量信息集的基本信息结构项，P(s|X) 为文书 X 中包含基本信息结构项，n 为模块数。

可读性

裁判文书是一种特殊的文本，它同时包含了司法特性和文本特性。裁判文书公开对象具有无差别性，即案件当事人可能为目不识丁者亦可能为高端知识分子。因为考虑到我国人口众多且知识水平参差不齐，所以评估裁判文书的可读性极具必要。在不降低法律文书水平的前提下，重视法律文书的实用性和通俗性、提高法律文书的说理性来增强法律文书的大众阅读性是法院法律文书公开的基本前提 [55]。

裁判文书主要结构固定，片段标注清晰较易区分，然而如原告诉称段、被告俗称段、查明事实段等为大段文本，因此衡量文书的可读性可选择“段落”的下一个单位——“句子”作为语言单位，通过依存句法分析计算句子平均依存距离。该过程仍需基于分词，分析标注当前词与关联词之间的句法依存关系。

多个研究 [56][57] 表明人类语言的普遍特征是依存距离最小化倾向。距离和认知负荷密切相关，两个词之间距离越大，认知负担就会越大，因此认为句子更为复杂。本指标采用了句子平均依存距离的计算公式 [58]:

$$Readability(X) = \frac{1}{n-1}\sum_{i=1}^{n} |DD_1| \tag{3-2}$$

其中，n 为句子的总词数，DD_1 为第 i 个句法连接的依存距离。

刘海涛 [57] 曾使用大规模语料文本对 20 种语言的依存距离进行研究，发现汉语的依存距离均值在 3.66。并且根据现代心理学普遍认定的工作记忆容量阈值为 4。根据以上公式及阈值可以判断该文书的可读性。

响应性

瑞典在 1997 年对法院行为质量展开了广阔的讨论，认为审理时间是法院行为重要的质量因素之一。高质量的一个重要标准就是尽快审理并判决案件。高质量的处理速度要求法官对程序的管理应当是高效的，案件的审理程序预先设

计且保持连贯。待相关事项准备就绪做出判决。对于待审案件必须按照时间先后顺序跟进，监控并约束审理时间。法院必须准备适当的诉讼计划。

本维度通过评估当前案件的响应性反映司法机关的工作效率，同时帮助评估诉讼的时间成本。响应性为自立案至开庭审理并判决所经历的时间，代表了司法机关法院能否及时处理解决案件，按期执行并完成审判工作。响应时间涉及到了人民法院任务的负荷和工作的规律性。响应时间也受当事人影响，即由于主客观原因耽误诉讼时间导致延期。

合理提高响应性，可以避免拖拉作风以及加强对当事人的约束，最终促进法院工作效率。

$$\text{Response}(X) = \frac{T_{pjsj} - T_{larq}}{T_{perfect}} \tag{3-3}$$

其中 X 为该篇文书信息集，T_{pjsj} 和 T_{larq} 分别为案件判决时间和案件立案日期，$T_{perfect}$ 为最佳时长（单位为天）。

民事案件（不包括立决案件）和按司法法典第 8 章第 4 节第 1 款规定顺序审理的争议起诉。民事案件和争议起诉的最佳审理期限为 4 个月。依据此最佳审理期限可以计算出阈值。

表 3-12: 最佳审判期限（分阶段）

流程	时长
登记传唤申请（立即）和熟悉内容	1 周
作出回复的截止期限	2 – 3 周，取决于案件质量
作出潜在声明的截止期限	2 – 3 周
预审	书面准备阶段结束后 2 个月内
主审	预审后 2 周内
判决	主审开始后 2 周内

延迟性

本维度从案件信息层面分析，延迟性为自案件发生到法院受理，最后得到判决结果所经历的时间。延迟性反映了案件的复杂性，不仅包含了案件经过时长，同时也涉及到法院审判任务响应性。通过文书信息评估案件的延迟性助于分析某类案件历经时长的规律。

从案件当事人角度出发，快速得到诉讼结果及诉讼期限非常重要。民事案件通常涉及人们的日常生活，包括家庭、生活、收入、工作、财产和安全等。自案件发生之时起，当事人的利益一直在遭受侵害，持久时间越长，损失越重。并且一个未决的案件容易占据人们的思想，消耗精力和资源，无暇估计生活中的其他事情。因此评估延迟性对于案件当事人具有必要性。

$$\mathrm{Delay}(X) = \frac{T_{pjsj} - T_{fsrq}}{T_{unit}} \tag{3-4}$$

其中，X 为该篇文书信息集，T_{pjsj} 为案件判决时间，T_{fsrq} 为案件发生日期，T_{unit} 为时间单位。

3.5.3 粗粒度分析

对文书集的数据数理分析除了具有数据层面的意义，同时具有司法特性的意义。从数据层面来讲，剖析数据集的数据结构，挖掘数据的内部信息，可以发现数据集是否适合被用来训练或测试模型，如验证数据扩增输出的新数据集的有效性，同时可以对使用过程或结果中存在的问题找到合理的解释。从司法特性来讲，裁判文书是具有特殊意义的最为宝贵的司法资源，通过大量的文书内容整合分析获得信息可对法律行业产生助力，帮助法官、律师、政府工作人员等人群从不同角度如地区、审理时间、判决倾向等分析决策或预测案情未来走向，有针对性的确定应对策略。

面对大量的文书，人工阅读分析并评价所耗费的时间无法估量。利用时代背景下的人工智能技术实现智能服务化可以协助人工快速了解数据集概况，与此同时也解决了人力资源浪费和时间浪费的问题。如今多数服务或辅助决策都基于数据，数据的覆盖性不足或内容缺陷等原因产生的数据偏差会导致决策偏差，因此数据质量和概况仍需人工把握。从事实角度和用户需求，本文从内容特征角度提供了几大指标，一般性统计、数据结构统计、异常值数据统计、争议焦点统计、地区案件统计。结合数据/信息质量的度量思想，提供了包括规范性、完备性、可读性、响应性、延迟性的统计分析。

事实

事实部分包括文书集文件的性质信息以及特征统计。性质信息为一般性统计信息，即文件集名称、文件数量、文件集大小、平均篇幅长度、篇幅长度标

准差。特征统计基于细粒度级别的事实统计信息包括单维度文书地区统计、判决时间统计、案件案由分布统计、审判法院层级统计、案件审理程序分布、案件来源分布、判决结果等，多维度如地区数量最高案由、地区判决结果、年份数量最高案由等，多维统计分析具有多种可能性，待进一步深入探索。为清晰且生动传达信息进而方便用户理解，本部分统计信息以可视化图表展示，如散点图、柱状图、饼状图等。

使用者通过文书集的数理分析可以推理获得多种结论，辅助决策。从“城市”角度分析，可以发现城市独有的诉讼，最容易发生离婚诉讼的城市、最勤劳的法院等多种有趣的信息。针对某类案由的地区判决结果统计，反映出法院在不同地区的审理特定案件时的倾向性，如美国的专利侵权诉讼通常在德州东区法院进行，因为该地区对专利权人的态度较为友好且审判高效，了解对自己诉讼请求更有利的管辖范围内的法院对于维权者来说非常有意义。

规范性

本部分依赖细粒度分析的规范性维度指标进而构建粗粒度分析的规范性维度指标。规范性指标为计数指标，以标点符号规范性为例，*batch_punc_norm* 为粗粒度分析的标点符号规范性指标，*punc_norm* 为细粒度分析的标点符号规范性指标。通过衡量数据集的规范性可有效评估数据质量，为使用者的数据处理方向或进一步工作提供参考依据。

$$\text{batch\_punc\_norm} = \sum \text{punc\_norm} \tag{3-5}$$

通过本维度检查数据集内各文书存在的规范性问题，存在规范性问题的文书将被收录至问题文书列表，以待进一步优化解决。

完备性

本部分依赖细粒度分析的完备性维度指标构建粗粒度分析的完备性指标。从粗粒度层面对所有文书的完备性统计，使用其平均水平反映文书集的完备性。该维度体现了数据集的必要数据是否完备，数据内容的缺失可能无法很好的代表真实世界情况。

丰富性

本维度指标帮助用户了解数据的丰富性。丰富的数据样本才可以反映出事务的客观规律。本文考虑司法特性，通过文书信息分析本数据集是否涵盖了所有民法案由以及是否涉及中国所有地区计算样本丰富度。根据《民事案件案由规定》计算案由丰富性，分别根据中国的省、市情况计算地区丰富性。

3.6 质量提升模块

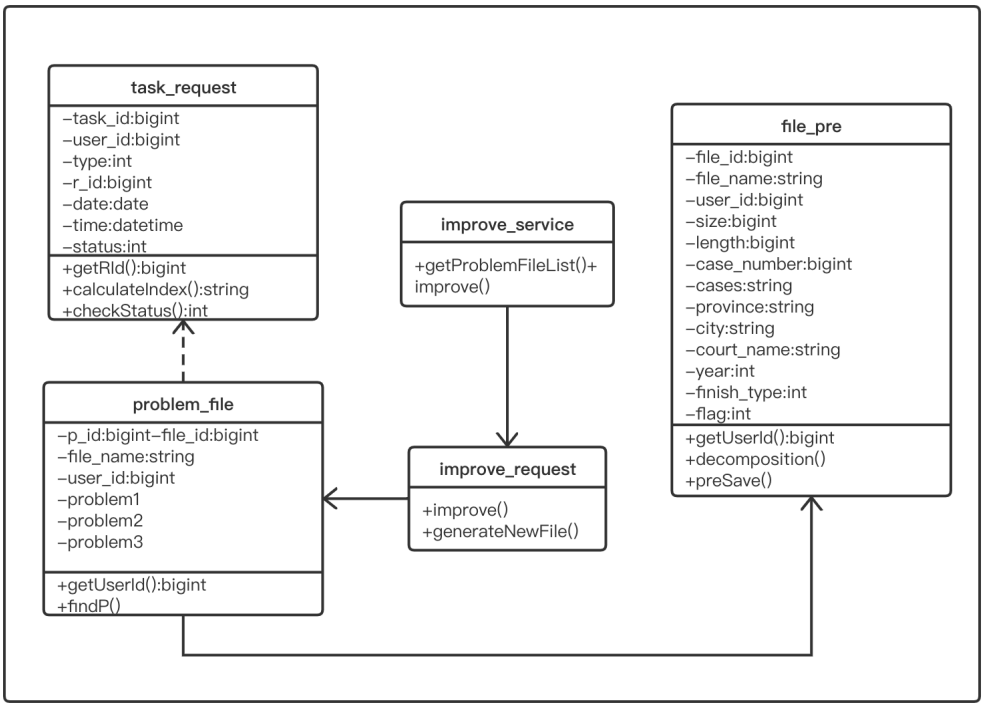

图 3-9: 质量提升模块设计类图

质量提升模块的运行依赖于分析任务产生的结果。*problem_file* 为问题文书相关信息包括问题定位信息，仅当用户至少执行成功一次分析任务后，可能检测获得问题文书。*improve_service* 负责质量提升服务，待用户在问题文书管理界面选中文书后，*improve_request* 获取该篇文书信息及问题定位信息，修复后形成新文书，存储至用户文件管理系统并与原文书关联。问题文书管理界面可以查看修复前和修复后的文件内容。

本部分通过裁判文书相关官方指导文件的解读映射以及对大量数据的统计分析总结问题参考列表，针对列表中的问题寻找解决途径，提供参考意见。用户根据系统检测所得文书的问题所在，判断是否需要进行修复，部分问题通过本系统的质量提升模块可自动化修复，属于第二类的质量问题则需人工进一步获取确认详细信息进行校正。问题参考列表如表 3-13所示，由于问题列表条目较多且支持扩展，此处仅展示部分。

表 3-13: 问题参考列表

问题编号	问题
pro_1	“本院认为”“被告辩称”等词语之后符号未使用逗号
pro_2	“××× 向本院提出诉讼请求”“本院认定如下” “判决如下”“裁定如下”等词语之后未使用冒号
pro_3	案号使用阿拉伯数字
pro_4	裁判尾部落款时间使用汉字数字
pro_5	裁判主文即列举阐述的序号使用汉字数字
pro_6	城市名称使用不规范，如“南京”应表述为“南京市”
pro_7	案由名称不规范
pro_8	缺失案由字段
pro_9	缺失当事人字段或起诉方、应诉方字段
pro_10	缺失诉讼费承担信息

3.7 数据库实体设计

3.7.1 结构设计

为保障系统有效运行，防止数据丢失，使用过程中的多个步骤及过程产生的信息记录均需持久化存储。相比于非关系型数据库，关系型数据库易于理解。考虑到使用过程中容易出现数据不一致的问题，本系统使用关系型数据库存储文书、用户等重要信息。数据库实体关系图为图 3-10。系统中需要进行持久化的数据主要涉及八个实体，其中 user 为系统用户表，upload_record 表记录用户上传文件请求，file_pre 代表用户拥有的文件及预处理信息，task 代表分析任务，report 为分析任务产生的分析报告，包括了细粒度分析报告 report_single

和粗粒度分析报告 report_batch，problem_file 代表问题文书的存储结构。

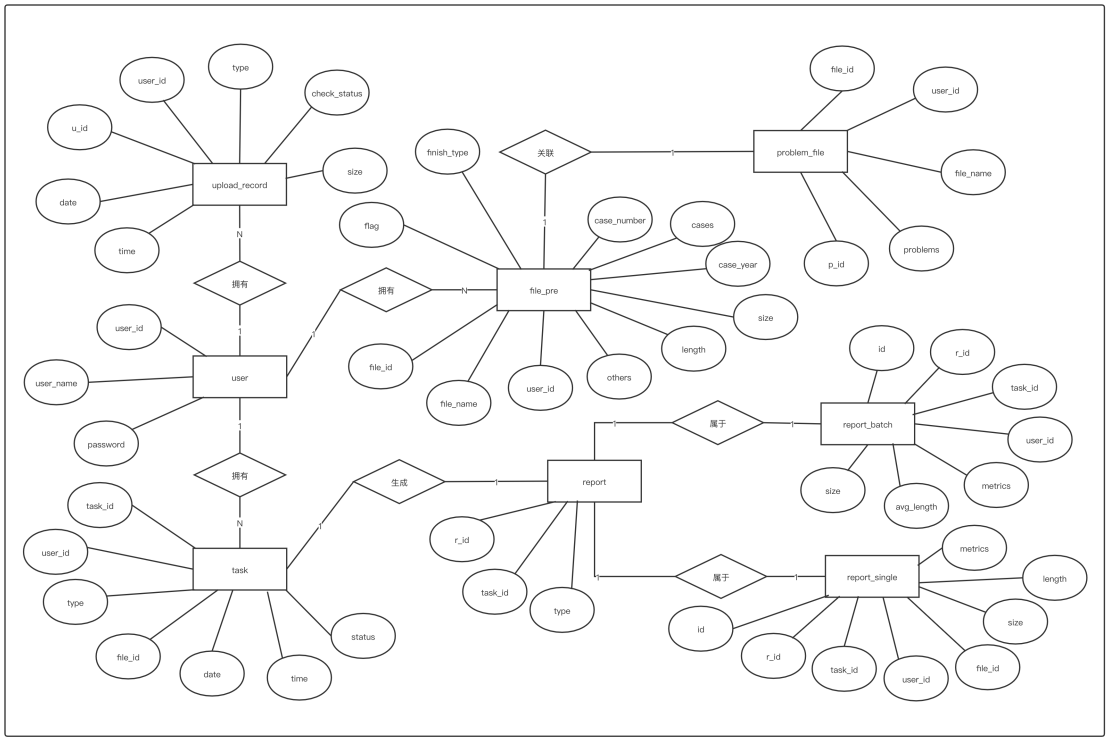

图 3-10: 数据库 E-R 图

表 3-14: user 表

字段	类型	含义	备注
user_id	bigint	用户 id	用户认证唯一标识
user_name	varchar	用户名	用户自定义昵称，要求唯一
password	varchar	用户密码	由数字、大小写字母、特殊符号组成，长度在 6-10 位数之间

登陆的用户即存在于 user 表的用户可以请求数据上传，用户可以多次上传数据，所以 user 与 upload_file 是一对多的关系。用户上传的数据类型可以为单篇文书的 xml 格式，也可以多篇文书打包的 zip 或 rar 格式。对于上传数据请求成功的数据，系统自动读取文件信息且针对文件格式自动进行预处理和基于裁判文书特性的结构解析，初步分析文件属性，将文件大小、长度以及案由、判决结果等信息存储于 file_pre 中，并且为是否为问题文书预留标记位置。用户选择待分析文书和分析任务类型后启动分析任务，一个用户可以创建多个分析

任务，即 user 与 task 是一对多的拥有关系，任务记录了单个任务的任务类型 type，与 report 实体相关联。一个任务只能且必须产生一个分析报告。由于不同粒度报告的度量维度不同，因此设立 report_single 实体与 report_batch 实体与 report 关联。此处分析报告实体关系的设计既保障了分析报告的唯一性，也体现了“低耦合”的思想。problem 实体定位了问题文书的异常，方便修复。

3.7.2 数据库表设计

表 3-15: file_pre 表

字段	类型	含义	备注
file_id	bigint	文件 id	文书唯一标识
file_name	varchar	文件名	上传的文件名
user_id	bigint	用户 id	用户认证唯一标识
length	bigint	文书长度	文书字符数
size	bigint	上传数据大小	单位为 M
case_number	varchar	案号	由立案年度、法院简称、字号、案号顺序号组成
cases	varchar	案由	参考《民事案件案由规定》
case_year	int	立案年度	本案立案年度
province	varchar	省份	法院所在省份
city	varchar	城市	若省份为直辖市，此字段为区
court_name	varchar	法院名称	标准法院名称
court_level	varchar	法院级别	如基层、中级
source	varchar	案件来源	八大来源
procedure	varchar	适用程序	三类程序
finish_type	varchar	判决结果	0: 判决；1: 撤诉；2: 驳回上诉
flag	int	是否为问题文书	0: 否；1: 是

表3-14为用户表（user）的数据库字段说明。本表记录了具有使用本系统权限的用户的账号信息，包括了作为系统唯一标识的用户 id，用户名以及密码。要求用户名在本表中同样唯一，用户登陆时系统以用户名作为标识符，验证密码。系统预先为本表中的所有用户设置使用权限。

表3-16为上传记录表（upload_record）的数据库字段说明。已登陆的用户使用本系统核心功能的前提是数据已上传。本表记录了用户的上传文件记录，包括文件名称、日期、时间、文件大小、文件格式类型。本表预留了请求的状态字段，用于反馈该请求的执行情况，0 为失败，1 为成功，初始状况为 0。

表 3-16: upload_record 表

字段	类型	含义	备注
u_id	bigint	上传数据请求编号	上传请求唯一标识
name	varchar	文件名称	不支持特殊符号
user_id	bigint	用户 id	用户认证唯一标识
date	date	日期	格式如 20200303
time	datetime	请求创建时间	格式如 20200303 18:30:30
size	bigint	上传数据大小	单位为 M
type	int	文件格式类型	0:xml；1:zip；2:rar
check_status	int	请求状态	0: 失败；1: 成功

表 3-17: task 表

字段	类型	含义	备注
task_id	bigint	任务 id	任务唯一标识
user_id	bigint	用户 id	用户认证唯一标识
type	int	任务类型	0: 细粒度分析任务； 1: 粗粒度分析任务
file_id_list	varchar	文件 ID 列表	参与任务的文件 ID
r_id	bigint	报告 id	分析报告唯一标识
date	date	日期	格式如 20200303
time	datetime	请求创建时间	格式如 20200303 18:30:30
status	int	任务执行状态	0: 等待；1: 进行中；2: 完成； 3: 失败。初始状态为 0

表3-15为文件预信息表（file_pre）的数据库字段说明。本模型为事实模型。系统对于所有上传格式的文件需进行处理，以统一格式存储，方便用户进行数据管理。并且对于存储于用户文件空间中的所有文书，在预处理过程中

提取部分属性信息持久化存储待后续工作使用，减少分析任务的工作量。其中包括文书的部分内容特征，如案号、案由、年份、地区、法院名称、法院级别、案件来源、使用程序、判决结果等，为案件基本情况、裁判分析、法律法条引用预留字段，待后续细粒度分析任务完成后填充。预留了 flag 为问题文书标记，初始状态为 0，即正常文书。如分析任务过程中系统检测到该文书有异常，则反馈至本处，flag 值为 1。

表 3-18: report 表

字段	类型	含义	备注
r_id	bigint	报告 id	分析报告唯一标识
task_id	bigint	任务 id	任务唯一标识
type	int	任务类型	0: 细粒度分析任务； 1: 粗粒度分析任务
status	int	报告生成状态	0: 不存在；1: 存在

表 3-19: report_single 表

字段	类型	含义	备注
id	bigint	细粒度分析报告 id	细粒度分析报告唯一标识
r_id	bigint	报告 id	分析报告唯一标识
task_id	bigint	任务 id	任务唯一标识
user_id	bigint	用户 id	用户认证唯一标识
file_id	bigint	文件 id	文书唯一标识
length	bigint	文书长度	文书字符数
size	bigint	上传数据大小	单位为 M
metric_1	varchar	指标 1	参考细粒度分析指标
metric_2	varchar	指标 2	参考细粒度分析指标
metric_3	varchar	指标 3	参考细粒度分析指标

表3-17为任务表（task）的数据库字段说明。当用户符合“已登陆”、“成功上传文件”、“确定分析任务类型及度量维度”等条件后方可启动分析任务。每个任务自动分配 task_id 为本次任务的标识，记录任务的类型信息 type，触发的 report 表的报告唯一标识 r_id，以及本次任务的启动日期和时间。status 为本次

任务的状态，0 代表任务在队列中等待执行，1 代表任务已经开始执行且处于执行中的状态，2 代表任务执行成功，3 代表任务执行失败。初始状态为等待。

表 3-20: report_batch 表

字段	类型	含义	备注
id	bigint	粗粒度分析报告 id	粗粒度分析报告唯一标识
r_id	bigint	报告 id	分析报告唯一标识
task_id	bigint	任务 id	任务唯一标识
user_id	bigint	用户 id	用户认证唯一标识
avg_length	bigint	文书长度	文书字符数
size	bigint	上传数据大小	单位为 M
metric_1	varchar	指标 1	参考粗粒度分析指标
metric_2	varchar	指标 2	参考粗粒度分析指标
metric_3	varchar	指标 3	参考粗粒度分析指标

表 3-21: problem_file 表

字段	类型	含义	备注
p_id	bigint	问题文书 id	问题文书唯一标识
file_id	bigint	文件 id	文书唯一标识
file_name	bigint	文书名	上传的文件名
user_id	bigint	用户 id	用户认证唯一标识
p_1	varchar	问题 1	参考质量提升模块
l_1	varchar	定位	问题所在位置即标签
o_1	varchar	详细内容	问题的细节描述
s_1	varchar	解决办法	问题的修复办法
p_2	varchar	问题 2	参考质量提升模块
l_2	varchart	定位	问题所在位置即标签
o_2	varchar	详细内容	问题的细节描述
s_2	varchar	解决办法	问题的修复办法

表3-18为分析报告表（report）的数据库字段说明。每个分析任务必对应 report 表的一个 r_id。report 的每条记录包括 task_id，分析任务类型 type，以及

报告生成状态，0 代表报告不存在，1 代表报告已生成。如果 status 为 1，则根据 type 的值 0 或 1 分别至 report_single 或 report_batch 中查找报告的详细结果。

表 3-19为细粒度分析报告表（report_single）的数据库字段说明。本表记录了用户亟待得知的文书分析后的各维度度量评估结果。细粒度分析主要从规范性、完备性等维度出发，计算了包括数字规范性、案号规范性等指标信息，帮助用户了解单篇文书概况以及核心特征信息。

表 3-20为粗粒度分析报告表（report_single）的数据库字段说明。粗粒度分析更侧重于统计分析，并从数据/信息质量层面对数据集的规范性、完备性、丰富度等多个维度进行评估，协助司法研究者、数据测试人员等人群快速了解数据集概况，有针对性地处理问题数据，对于进一步的分析作出合理的解释。

表 3-21为问题文书表（problem_file）的数据库字段说明。本系统的一个特色为帮助用户定位问题数据，并在一定程度上进行质量提升。质量提升技术基于各大文书质量评估框架，专家建议，《样式》以及大量数据统计分析总结所得的多个问题，定位并记录文书异常位置。

3.8 本章小结

本章为项目的需求分析和概要设计。通过涉众分析进一步进行需求分析，从而完成系统用例分析。根据前序分析设计了系统的总体架构，并通过 4+1 视图和模块解析从面向不同对象及功能/非功能性需求各个角度描述构建了一个完整的系统。使用设计类图介绍了每个模块的构造，在度量解析模块详细介绍了细粒度分析和粗粒度分析的事实模型和多维度量体系的构建。最后，设计了数据库实体，包括了数据库结构设计及各个数据模型的设计。

第四章　详细设计与实现

本文将系统主要分为三个模块，数据交互模块、度量解析模块、质量提升模块。根据第三章描述的需求分析和概要设计，本章通过模块顺序图、关键代码等方式对这三个模块的详细设计与实现展开介绍。

4.1　数据交互模块

4.1.1　详细设计

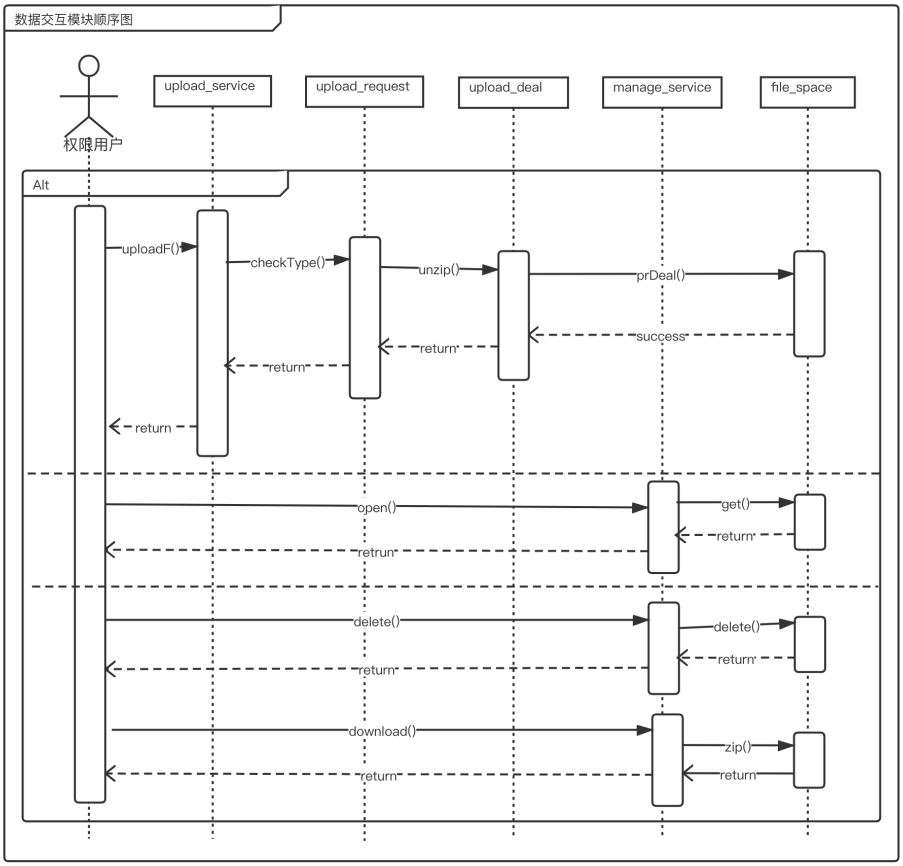

图 4-1: 数据交互模块顺序图

数据交互模块主要实现用户使用系统时文件的上传、下载及管理的功能。如图 4.1 为数据交互模块顺序图，记录了各个函数之间的调用顺序。已被系统识别的且拥有权限的用户触发数据上传服务 *upload_service*，根据系统指导，确定即将上传文件的路径，向系统发送上传请求。系统调用 *checkType*() 检查文件格式类型，无误则将文件保存至指定文件系统的用户目录下，如果文件为压缩文件则调用 *unzip*() 函数解压再保存。*upload_deal* 同时获取文件列表信息，调用 *pre_deal*() 函数对用户目录下的本次上传文件进行预处理，获得每篇文书的预处理信息并持久化存储。除上传功能外，用户在数据交互模块可进行文件管理，包括查看、删除、下载。此处展示数据主界面及上传界面图，如图 4-2所示。

图 4-2: 数据主界面及上传界面

4.1.2 关键代码

```
def pre_deal(upload_file_list,user_path):
    for i in upload_file_list:
        # 文书路径
        path=user_path+i
        dom = xml.dom.minidom.parse(path)
        # 获取文档元素对象
        data = dom.documentElement
        qw = data.getElementsByTagName("QW")[0]
        # 获取事实模型
        case_number=getAH(qw)
        case=getAY(qw)
        case_year=getYear(qw)
        ......
    return
```

本部分展示的代码为文书上传后的预处理过程。由于文书的许多信息在后续的任务中涉及到重复调用，为防止重复计算，将事实模型构建置于本阶段

完成。首先调用 xml.dom 解析裁判文书，用户上传文件成功后触发构建事实模型，读取文档元素对象，调用多个指标计算方法获取各项信息，如 *getAH* 方法获取文书案号，*getAY* 方法获取案由，同时将结果持久化存储。

4.2 度量解析模块

4.2.1 详细设计

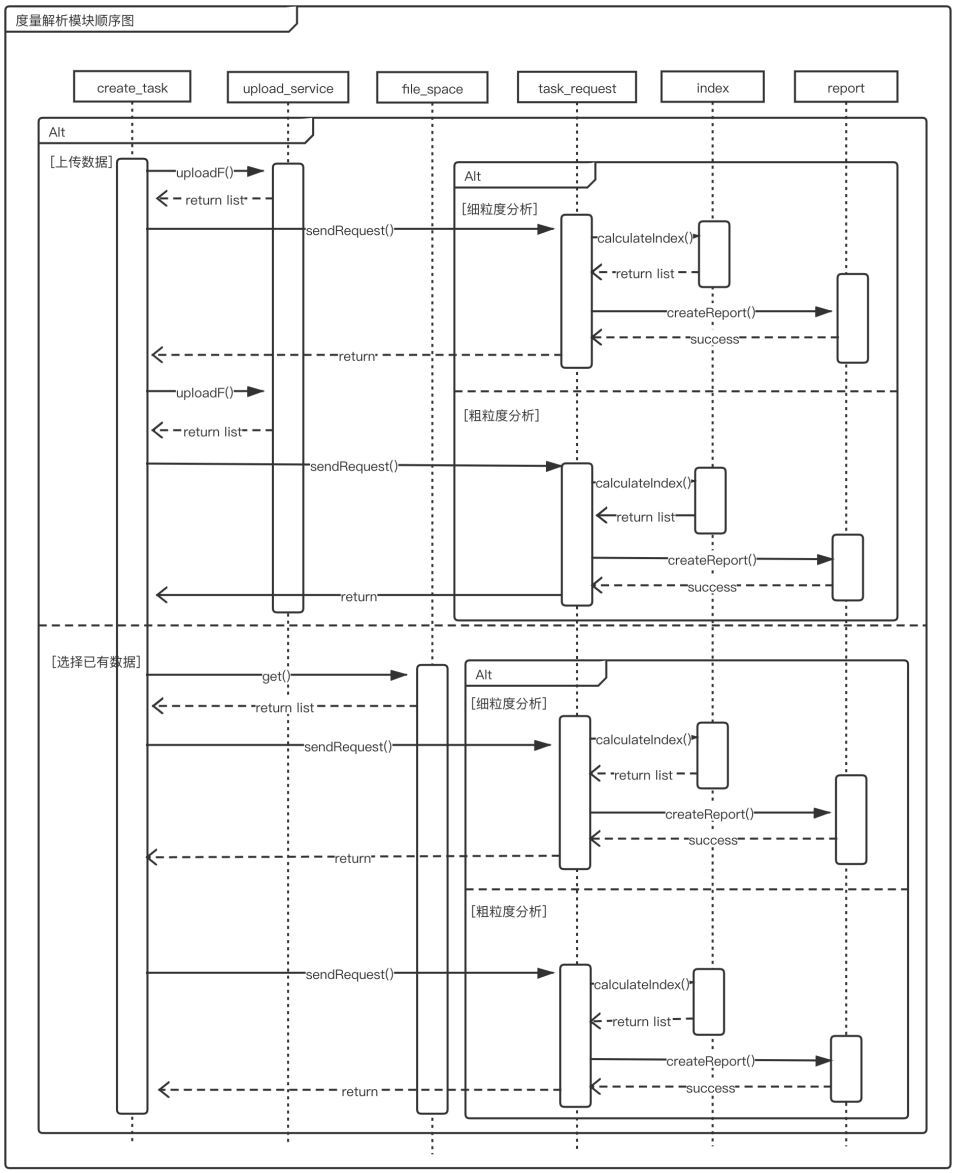

图 4-3: 度量解析模块顺序图

图 4-4: 任务管理界面

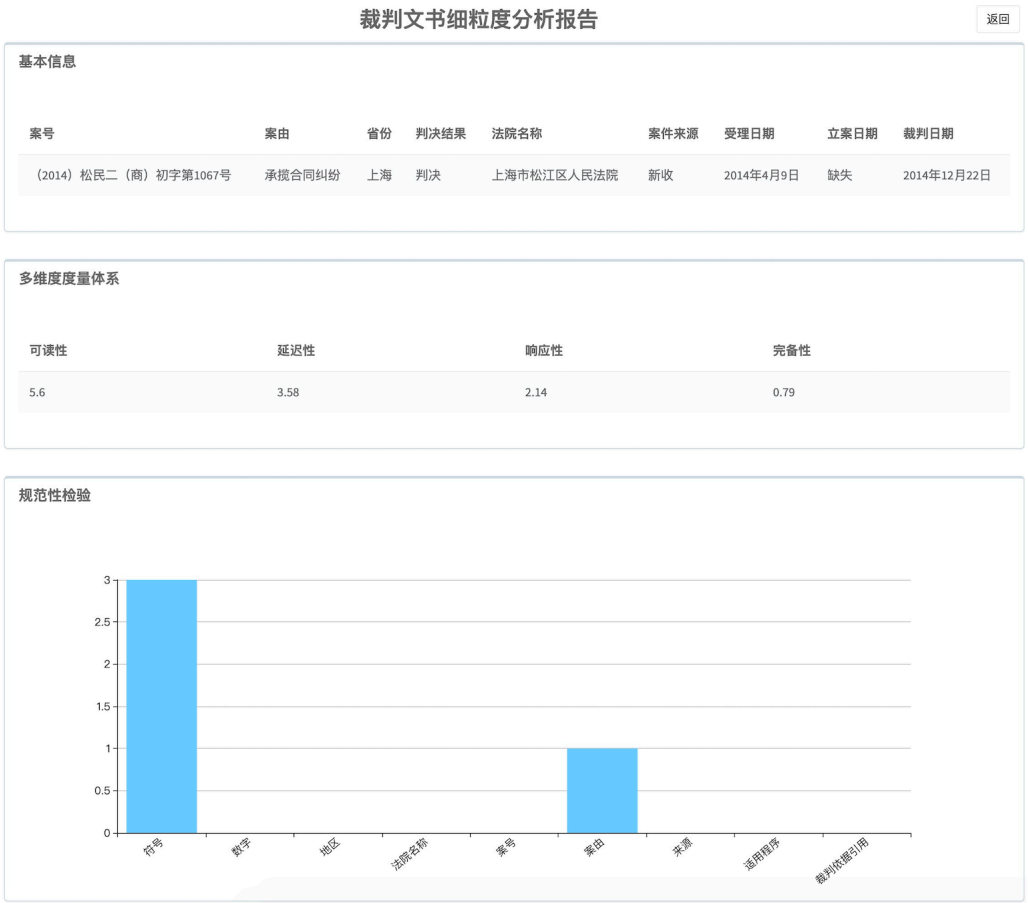

图 4-5: 细粒度分析报告（一）

度量解析模块的主要功能是通过预先设计的规则完成文书解析和度量体系的指标计算及融合，该模块为本系统最核心的部分。每篇文书拥有其特有的事

实模型及多维度量指标体系，数据集的度量解析建立在单篇文书的度量解析上，融入统计特征，同样具有事实模型和多维度量指标体系。本模块面向单篇和多篇文书展开分析，根据文书及数据集特性设计多维度量体系，根据不同的操作呈现可表征该数据或数据集信息的分析报告。如图4-3所示为文书或文书集创建并启动分析任务时对各个部分的调用顺序。

图 4-6: 细粒度分析报告（二）

图 4-7: 粗粒度分析报告（一）

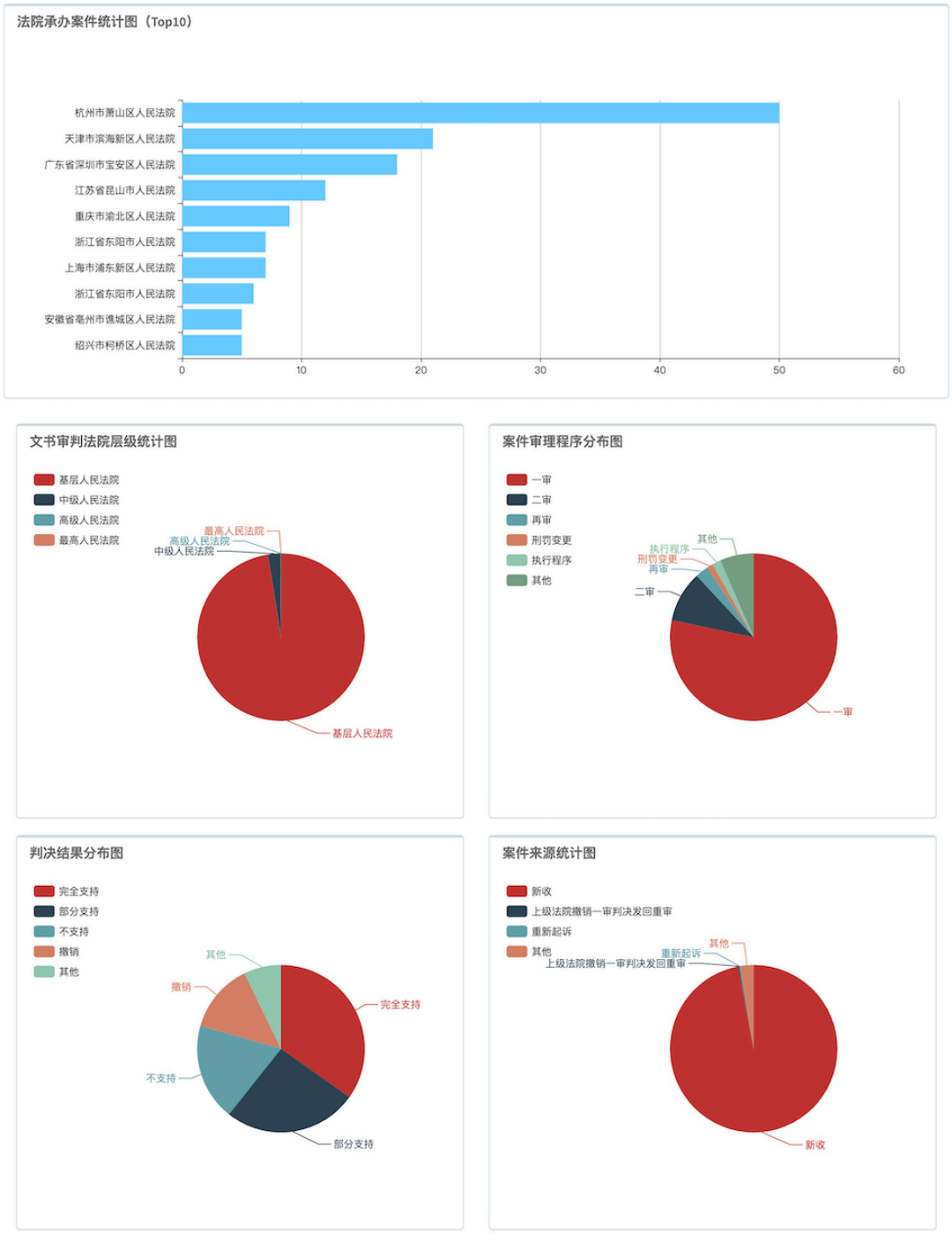

图 4-8: 粗粒度分析报告（二）

用户创建分析任务时 *create_task* 触发文件选择界面，用户可以选择立即上传系统调用 *upload_service* 类，过程与数据交互模块的上传文件部分一致；或者选择已有数据，则调用 *file_space* 类并将用户选择文件 id 以列表形式返回。

create_task 获得待测文件后根据用户选择的任务类型及度量指标发送请求启动细粒度分析或粗粒度分析任务。*task_request* 类根据前序传送参数配置任务启动分析，进入 *Index* 调用具体指标方法计算结果。*task_request* 创建 *report* 对象，并将具体数据持久化至数据库。*create_task* 最终获得成功反馈信息，任务结束。待任务成功后，用户可选择在线查看或下载分析报告。该动作触发系统 *reportGeneration* 类自动对 report 对象信息布局展示，生成分析报告，展示给用户。此步骤为本系统及用户行为的最终目的。

本模块涉及功能和实现界面较多，此处仅展示部分核心界面。图 4-4为任务管理界面，用户启动细粒度或粗粒度分析任务后，点击左侧的任务列表进入该界面。根据任务命名获得任务状态，如任务状态显示为“查看报告”，则允许点击，进入分析报告界面。分析报告根据任务类型同样分为细粒度分析报告和粗粒度分析报告，由于报告篇幅过长，此处仅截取部分信息进行展示，图 4-5和图 4-6属于细粒度分析报告，图 4-7和图 4-8属于粗粒度分析报告。

4.2.2 关键代码

度量解析模块的关键点在于支持多类型任务分析及多维度量指标体系建立，后者的难点更侧重于应用文书相关的司法知识建立规则体系，此处不多赘述。而技术实现难点仅展示细粒度分析任务，案情摘要。

4.2.2.1 细粒度分析任务

对于细粒度分析任务，每次任务仅针对单篇文书进行解析并获得报告，粗粒度分析任务则针对数据集做数理统计分析。为增加系统容错性，每次任务数据以列表形式输入，当用户上传多篇文书且启动细粒度分析任务，仅读取首篇文书执行任务。以细粒度分析为例，代码如图 4-9所示。首先读取文件内容，然后使用多维度量体系进行文本的质量分析。规范性方面，调用 *con_pun* 方法进行符号规范性度量，调用 *numMethod* 方法进行数字规范性度量，调用 *locationMethod* 方法进行地区规范性度量，调用 *courtNameMethod* 方法进行法院名称规范性度量，调用 *casenumMethod* 方法进行案号规范性度量，调用 *aut_AY* 方法进行案由规范性度量，调用 *sourceMethod* 方法进行来源规范性度量，调用 *procMethod* 方法进行适用程序规范性度量，调用 *relyMethod* 方法进行裁判依据引用规范性度量；然后调用 *completenessMethod* 方法进行完备性度量；调用 *readabilityMethod* 方法进行可读性度量；调用 *timeMethod* 方法

进行延迟性和响应性度量；调用 *timelineMethod* 方法进行时间线抽取；调用 *abstractMethod* 方法进行摘文本摘要生成。

```
def analyze_single(report, file):
    # 省略获取文本内容
    punc_count = con_pun(qw_content)
    num_count = numMethod(qw)
    loc_count = locationMethod()
    court_count = courtNameMethod(qw)
    case_count = casenumMethod(qw)
    ay_count = aut_AY()
    source_count = sourceMethod(qw)
    proc_count = procMethod(qw)
    rely_count = relyMethod(qw, )
    completeness = completenessMethod(file.id, qw)
    readability = ReadabilityMeasure.readabilityMethod(qw_content)
    response, delay = timeMethod(qw)
    timeline = timelineMethod(qw)
    abstract = abstractMethod(qw)
```

图 4-9: 细粒度分析任务关键代码

4.2.2.2 案情摘要

```
def predict(text, topk=3):
    # 抽取
    texts = convert.text_split(text)
    vecs = vectorize.predict(texts)
    preds = extract.model.predict(vecs[None])[0, :, 0]
    preds = np.where(preds > extract.threshold)[0]
    summary = ''.join([texts[i] for i in preds])
    # 生成
    summary = seq2seq.autosummary.generate(summary, topk=topk)
    # 返回
    return summary
```

图 4-10: 文本摘要关键代码

为减少用户的阅读时间，本文提供案情摘要即尽量以精简的文段陈述案情经过及法官说理。自动化文本摘要使用原文书内容，通过算法训练生成有限字数的文本且几乎表达原文核心信息。

为获取高质量可用摘要，本文参考 SPACES[59] 采取“抽取 + 生成”相结合的方式进行摘要。抽取模型部分为使用规则将原始的生成式语料转

化为序列标注式语料，使用自行构建分句函数 *text_split*() 使得句子的颗粒度更细，调用 *vectorize.predict*() 方法对摘要的每个句子都在原文中匹配与之相似度最高的那个句子。将所有匹配到的原文句子作为抽取句子标签，调用 *extract.model.predict*() 训练模型，设定阈值 *extract.threshold* 抽取摘要作为抽取模型的输出。由于抽取模型的目的是为了生成模型获取高质量摘要，在抽取过程中应尽量覆盖到最终摘要所需的全部信息，调用 *seq2seq.autosummary.generate*() 将抽取结果通过 Seq2Seq 模型优化生成最终摘要。本文的生成模型使用了中国法研杯的司法摘要数据进行训练。

4.3 质量提升模块

4.3.1 详细设计

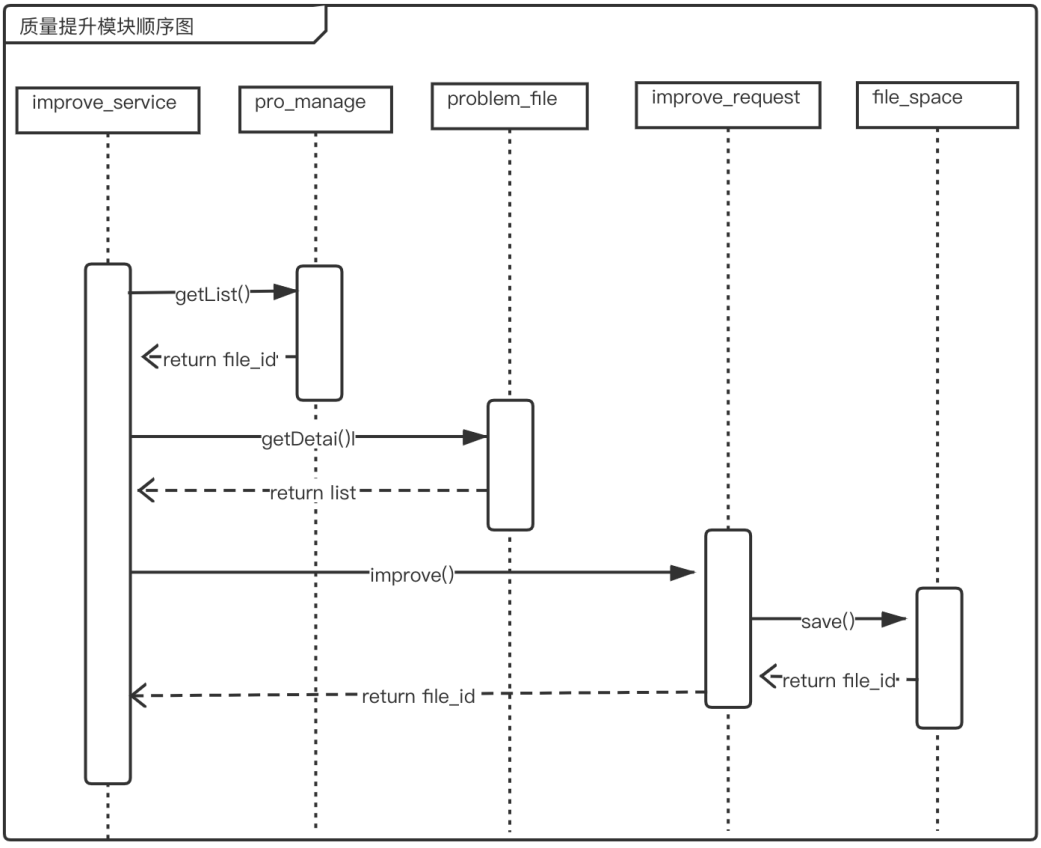

图 4-11: 质量提升模块顺序图

数据质量问题主要分为由于常识或规定等已知信息可以定位的问题以及需要多方的输入确认藏于细节中的问题。本系统仅针对第一类问题，根据现有规则及文本信息完成修复。第二类问题依赖现实情况或专业知识，本文仅定位错误。尽管如此，Jim Barker 认为第一类问题占据了所有问题的 80%，解决该部分问题足以在一定程度上提升数据质量。

图 4-12: 问题文书管理界面

图 4-13: 文书问题列表

质量提升模块的目的是帮助用户解决数据的规范问题及部分内容缺失问题。图 4-11为质量提升模块顺序图，包括用户对问题文书管理及选择问题文书一键提升质量。*improve_service* 类是本部分的入口，系统触发 *getList*() 方法进入 *pro_manage* 类获取问题文书列表，用户通过查看文书内容确认是否需要修复，无需修复可选择删除，被删除文书恢复为正常文书，仅从问题文书列表中删除，依旧可以在数据交互模块的 *manage_service* 类对该文件进行查看、删除、下载等操作。用户在问题文书列表中选择待修复文书，*pro_manage* 类返回该文书 *file_id*。*improve_service* 根据获取的 *file_id* 向 *problem_file* 获取问题定位，并调用 *improve_request* 对定位问题进行修复生成新文书，存储至 *file_space* 并与原文书关联，返回新文书的 *file_id*。用户可在数据交互模块的文书管理处下载修复后的新文书。

分析任务的启动会触发问题文书的检测，检测后的文书会出现在问题文书管理列表中，如图 4-12所示，除了展示该文书的基本信息外，提供“查看问题”

和“修复”按钮，点击第一个按钮可查看该篇文书的问题列表，如图4-13所示，展示了对应的多个问题，以及每个问题的涉及标签、原文描述和修改建议。

4.3.2 关键代码

```
def detect_pro(file_path):
    # 获取问题列表
    pro_list=get_problem()
    pro_length=len(pro_list)
    # 问题是否存在
    pro_index=['0' for n in range(pro_length]
    dom=xml.dom.minidom.parse(filr_path)
    data=dom.documentElementsByTagName("QW")[0]
    list_all=[]
    # 对每个可能存在的问题进行检测
    # 返回 list，是否存在，定位，错误内容，更新内容
    list_1=pro_1(qw)
    list_all.append(list_1)
    # 此处省略多个方法
    for i,j in enumerate(list_all):
        if j[0]==1:
            pro_index[i]=1
    return pro_index
```

图 4-14: 问题文书检测关键代码

识别问题文书部分主要针对民事裁判文书结构判断要素缺失以及数据规范性问题。细粒度分析任务的多维度量体系中的规范性维度和完备性维度与数据质量息息相关，针对这两个维度的多个指标，本系统根据具体规则采用不同的识别方法。标点符号及数字等规范性则使用正则表达式完成识别，案由则使用案由词典判断其规范性，案号的规范性检测则检测其是否符合固有格式。无论用户启动的是细粒度分析还是粗粒度分析任务，在任务执行过程中的均会对待测文书进行问题识别，并生成其问题详情列表，其中包含问题描述，涉及标签，原文描述以及修改建议。

本部分提供人性化的操作选择，将问题文书归类到问题文书管理部分，同时提供了问题详情列表及自动化修复功能，用户可以查看文书问题详情列表进而决定是否修复。修复时，系统根据问题文书列表，读取定位和错误信息，调用修复机制，对错误信息进行覆盖，并生成新文书，防止系统修复效果未达到用户需求的同时导致原文件丢失。关键代码如图4-14所示，调用 *get_problem*()

方法获取预设问题列表，根据预设问题列表检测，分别调用 *pro*_1 等各个问题的检测方法，判断是否存在该问题，并返回错误内容等信息。同时每个问题检测方法中会调用相应的修复方法，返回修复建议。

4.4 本章小结

本章描述了可解释性分析技术中各个模块的实现细节。通过模块的顺序图介绍了实现过程。对于度量解析模块的任务选择、案情摘要、案情时间线梳理等多个核心技术重点使用了关键代码进行了详细阐述，并展示了分析任务报告。在质量提升模块，主要描述了问题文书检测及文书修复机制的实现。

第五章　系统测试与分析

5.1　功能测试

本小节的测试目标是检测数据质量驱动的裁判文书可解释性分析技术可以满足用户需求，提供相应服务，具备一定的可用性。根据前序论文描述的内容，分别对系统的各个模块功能进行测试评估，验证系统的可靠性和性能。

5.1.1　测试设计

根据第三章系统需求分析得到的功能性需求，本小节进行功能测试测试用例的设计，面向系统的功能进行测试，目标是保证数据质量驱动的裁判文书可解释性分析技术能够提供预期的功能，满足用户的业务需求，进而达到功能覆盖的标准。

表 5-1: 数据上传交互测试用例

用例编号	TC1
测试名称	数据上传交互测试
前置条件	用户已经得到授权和认证
正常流程	1. 点击“选择文件”按钮 2. 用户选择想要上传的文件 3. 点击“上传”按钮
预期结果	1. 展示文件选择页面，只包括符合格式要求的文件 2. 预览该文件 3. 上传文件，进行初步处理，返回成功上传信息 4. 跳转至文书管理界面

表5-1展示了数据上传交互测试用例，测试的是用户上传文件的功能，主要关注点是用户能否正常上传符合要求的文书数据集，需要测试的具体功能包括文件上传，文件预览，文件格式检测。测试结果符合预期。

表5-2展示了文件管理测试用例，测试的是用户对于自己上传的文件进行管理的功能，主要测试的功能包括查看文件，删除文件，下载文件，多文件打包下载等。测试结果符合预期。

表 5-2: 文件管理测试用例

用例编号	TC2
测试名称	文件管理测试
前置条件	用户已经得到授权和认证
正常流程	1. 点击“个人文件管理”按钮 2. 点击文件“查看”按钮 3. 点击文件“删除”按钮 4. 选择文件，点击页面上方“下载”按钮 5. 搜索框中输入文书名称
预期结果	1. 展示用户上传的文件列表 2. 展示文件详细内容 3. 删除文件，提示“删除成功” 4. 将所选文件打包下载 5. 文书列表仅展示该文书

表 5-3: 启动细/粗分析任务测试用例

用例编号	TC3
测试名称	启动细/粗粒度分析任务测试
前置条件	用户已经得到授权和认证，至少已上传一篇文件
正常流程	1. 点击“细粒度解析”或“粗粒度解析”按钮 2. 选择数据源 3. 选择细/粗粒度指标 4. 点击“启动”按钮
预期结果	1. 展示细/粗粒度分析参数配置界面 2. 启动细/粗粒度分析任务 3. 提示“创建成功，等待报告” 4. 生成细/粗粒度分析报告

表 5-4: 分析报告交互测试用例

用例编号	TC4
测试名称	分析报告交互测试
前置条件	用户已经得到授权和认证，启动过分析任务
正常流程	1. 进入任务管理界面 2. 点击“刷新”按钮 3. 点击“查看报告”按钮 4. 点击“报告下载”按钮
预期结果	1. 展示分析任务进度，通过刷新页面完成“等待”、“进行中”、“完成”等多种任务状态的转换 2. 展示分析报告详情 3. 下载分析报告至本地

表5-3展示了启动不同粒度分析任务测试用例，测试用户启动文书分析任务的功能是否有效执行，主要关注点是粗细粒度不同的度量指标和维度。需要测试的具体功能包括启动细粒度分析任务，启动粗粒度分析任务，选择度量指标，跳转至任务管理界面，生成分析报告等。测试结果符合预期。

表 5-5: 问题文书管理测试用例

用例编号	TC5
测试名称	问题文书管理测试
前置条件	用户已经得到授权和认证，成功执行分析任务
正常流程	1. 点击“问题文书管理”按钮 2. 点击文书 A 的“查看问题”按钮查看问题信息 3. 点击文书 A 的“删除”按钮 4. 点击修复后的文书 B 的“查看内容”按钮 5. 点击文书 B 的“删除”按钮
预期结果	1. 展示问题文书列表 2. 展示文书 A 的问题列表 3. 删除无需修复的文书 4. 查看修复后的新文书 5. 删除不满意的新文书

表5-4展示了分析报告交互测试用例，测试的是用户管理分析报告的功能，主要测试的具体功能包括查看分析任务进度，包括等待、进行中、完成等多种状态的转换，在线浏览分析报告，下载分析报告等。测试结果符合预期。

表5-5展示了问题文书管理测试用例，测试的是用户对于问题文书的管理功能，主要关注点是对可疑文书的管理，测试的具体功能包括查看待修复文件，删除待修复文件，查看修复后文件等。测试结果符合预期。

表5-6展示了问题文书修复测试用例，测试的是用户在系统帮助下进行问题文书修复的功能，主要关注点是提升数据质量，发挥数据价值。用户确认修复建议符合要求后，点击按钮，系统自动化修复文书并生成与原文书关联的新文书。测试结果符合预期。

表 5-6: 问题文书修复测试用例

用例编号	TC6
测试名称	问题文书修复测试
前置条件	用户已经得到授权和认证，成功执行分析任务
正常流程	1. 点击“问题文书管理”按钮 2. 点击原始文书“修复”按钮 3. 点击新文书“查看内容”按钮
预期结果	1. 展示问题文书列表 2. 启动文书修复任务 3. 该篇文书的“修复”按钮更新为“查看内容” 4. 展示修复后的文书详情

5.2 接口测试

接口测试主要是针对服务与服务之间交互的 API 的测试，由于目前大部分软件开发过程中，系统内部的具体实现迭代很快，但对外提供的接口一旦确定往往不会更改，因此接口测试是系统测试中重要的一环。本节针对系统中较为重要的关键接口进行测试，保证接口的正确性和稳定性。

5.2.1 测试设计

本系统执行的接口测试用例如表5-7所示，主要针对系统的高优先级接口进行测试，考虑数据一致性和返回结果是否符合预期。本文采用接口测试工具JMeter进行接口测试，JMeter可以模拟HTTP请求，自定义请求URL、请求类型，提供了强大的接口调用与测试能力。

表 5-7: 接口测试用例表

测试编号	接口说明	输入	预期输出
I1	测试正常文件上传	符合格式要求的文件	上传成功信息
I2	获取文件列表加载	用户 ID	文件列表
I3	测试查看文书	正确文书 ID	文件内容
I4	测试删除文件	文书 ID	删除成功提示
I5	测试新建分析任务	文书 ID，分析指标	任务状态
I6	测试加载任务报告	任务 ID	报告 ID
I7	测试下载分析报告	任务 ID	分析报告
I8	测试问题详情列表	用户 ID	问题文书列表
I9	测试查看问题详情列表	文书 ID	文书问题列表
I10	测试删除问题文书	文书 ID	删除成功提示
I10	测试删除问题文书	文书 ID	删除成功提示
I11	测试修复问题文书	文书 ID	返回新文书 ID

本测试以编号I5为例，使用JMeter对接口进行测试，检测系统的并发性。具体步骤分为以下两部分。

1）图5-1为线程组具体配置，设置线程组中Number of Threads=100，Ramp-Up Period=10，Loop Count=5，代表线程数量即虚拟用户数量为1000，在10s内循环发送5次请求。

2）配置启动细粒度分析任务接口的HTTP请求，包括请求协议，IP，端口，请求类型，路径，编码，参数等。图5-2为具体配置内容。

非功能性需求在用户体验中占据着重要地位，根据需求设计中的非功能性需求解析，本部分通过模拟用户实际操作的系统执行任务的响应时间测试系统

的并发性以及时间特性，而其余特性如可用性、鲁棒性、安全性等均在项目设计中考虑到位并在实现过程中加以测试。

图 5-1: 线程组配置示例图

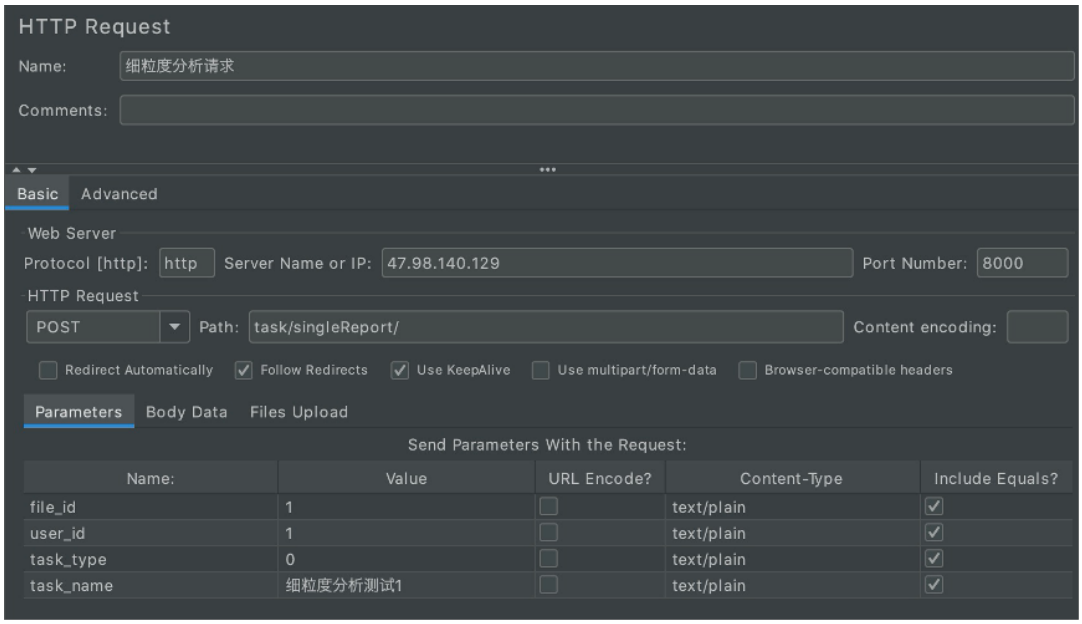

图 5-2: HTTP 请求配置示例图

5.2.2 测试结果

接口测试的结果如表5-7所示，所有接口在高并发情况下的测试均通过且响应时间较短，说明本系统的接口具有正确性和一定的稳定性。

启动细粒度分析任务接口是所有接口中使用频率最高、响应时间要求最高的接口之一。以该接口为例，由图5-3可知，100 个线程在 10 秒内各自发送了

5 次。启动细粒度分析任务请求，共 500 次请求，错误请求数为 0，最大和最小响应时间分别为 108ms 和 23ms，平均响应时间为 28ms。

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB...	Sent KB/sec	Avg. Bytes
细粒度分析请求	500	28	23	108	5.87	0.00%	50.3/sec	655.50	11.11	13336.0

图 5-3: 启动细粒度分析任务测试结果

表 5-8: 系统任务响应时间表

系统任务	平均值/ms	最大值/ms	最小值/ms	测试结果
文件上传	21	55	15	通过
文件列表加载	14	36	8	通过
查看文件	35	57	24	通过
删除文件	23	78	20	通过
新建分析任务	28	108	23	通过
加载分析报告	15	38	8	通过
下载分析报告	9	14	7	通过
问题文书列表加载	15	33	12	通过
查看问题详情列表	13	46	10	通过
删除问题文书	24	67	16	通过
修复问题文书	41	89	31	通过

5.3 本章小结

本章根据系统的需求分析针对功能性需求进行了测试，验证了系统的可用性，能够满足用户的业务需求，有一定的实用价值。使用接口测试对系统对外提供的接口进行了测试，保证了系统服务的稳定性。最后使用系统对真实的裁判文书进行修复，说明了数据质量驱动的裁判文书可解释性分析技术的有效性，可以提高数据质量，发挥数据价值。

第六章　总结与展望

6.1　总结

在司法公开不断深入的背景下，裁判文书数量以惊人的速度爆发式增长。裁判文书在司法审判中具有不可撼动的地位，充分利用裁判文书的信息和价值是智慧法院建设的一个发展方向。对于裁判文书的不同受众而言，可解释性分析技术都提供了高效获取文书及文书集信息以及多维度量体系作为参考，同时指出文书存在的问题帮助使用者提高文书数据质量。

面对长篇大论的裁判文书或包括数量众多文书的数据集，阅读者或数据使用者面临在获取其信息以及其多方面概况时如何提高效率的困扰，本文提出了可解释性分析技术。可解释性分析技术是本文的核心技术，构建了事实模型和多维度量体系对文书和文书集进行分析并生成分析报告。在文书事实提取部分调研了学术界和工业界的研究进展，包括关于裁判文书的组成要素以及各个要素的重要程度，文本摘要以及文本可视化等快速获取文本信息的技术，文本数据处理所涉及的自然语言处理技术等；而文书集的事实模型则使用数理统计的方法，通过不同维度的分布反映数据概况，通过大样本数据即符合大数定律可反映社会现象。在多维度量体系部分调研了数据质量的发展和已应用于裁判文书的质量评估体系，结合本技术的目的重新构建度量体系。

针对现阶段存在的裁判文书质量问题，司法机关不断呼吁深化改革裁判文书质量，本文提出的质量提升技术结合裁判文书制作规范等相关材料，在分析任务过程中通过问题检测功能筛选问题文书以及对应问题详情列表，提供对应问题的修复思路。对于无需人工进行细节信息核对问题可通过一键自动化修复，减轻使用者负担，轻松提升文书数据质量。对于另一类问题，则交给使用者去优化，系统仅提供定位信息和错误信息。

为方便用户使用，将上述技术集成为系统，提供用户友好的操作界面及引导，同时设置文书及问题文书管理部分，包括上传、查看、下载、删除等基础功能。以任务为单位执行分析获取报告，用户可对已启动的任务进行管理，查看任务进度，可以终止等待中的任务。对于已完成任务生成的任务报告，为用

户提供在线浏览及下载至本地两种服务。从系统开发的角度，设计整体架构图，通过 4+1 视图描述系统，确定功能性及非功能性需求，根据功能划分模块实现，并使用类图、过程图详细解析了模块组成及调用顺序。

最后，本文介绍了系统各个模块的具体实现，完成各个功能模块的测试分析，使用真实文书的指标计算结果对比系统计算结果以验证指标准确性，设计并使用测试用例确保系统的高可用性，包括功能性与非功能性需求的满足。

6.2 展望

本系统初步实现了调用事实模型及多维度量体系的各指标完成分析任务和问题文书的检测等功能，帮助用户了解单篇文书甚至文书集的信息和多维度概况，然而由于本技术涉及司法领域的专业知识，在细粒度分析和粗粒度分析层面从事实模型和多维度量体系上仍具有较大的优化空间。

细粒度分析的多维度量体系除已确定的维度，还可以从以下多个维度进行度量解析。从司法本身具有的公平正义的角度，度量体系可以对文本进行情感分析度量，判断法官是否存在情感倾向性。结合语法语义，从说理的充分性、论证的到位度方面度量释法说理的合理性，从而判断文书定分止争的效果。

在粗粒度分析上，目前仅提供部分常用特征分析，不够全面。由于不同用户对数据集的具体需求大不相同，整体而言由于用户需求不明确，可根据用户需求提供更大的操作空间。后续可以尝试提供接口，用户通过 SQL 语言对数据进行操作后，调用可视化图表接口获得统计结果。考虑到部分使用人员不具备数据分析的技能，需提供更加友好的界面供用户操作。

另外，结合智慧法院建设中现有技术的优点及弊端优化质量提升模块，同时进行反哺。裁判文书自动生成技术现已投入应用，然而仍未达到理想效果，问题文书检测及质量提升部分后续的优化方向可以结合裁判文书自动生成规则进行完善并且使用深度学习技术自动识别，从而协助自动生成技术的优化。

致谢，应放在结论之后

致　　谢

16 年初，与友同游南京，金陵古韵，婉转悠扬，爱上这座城市。

18 年 7 月，幸至南大参营，一砖一瓦，一草一木，放映着百年故事，爱上这所学校。

一晃二载有余，恍然如昨日刚踏入南大，转瞬将别，踏上新征程。在南大的求学生活充实而短暂。

得遇良师，何其有幸。感谢陈振宇教授为我提供诸多学习实践的机会，得以参与多项活动和项目，开阔视野，提升能力。在从选题到设计、修改与定稿整个完成毕设过程中，帮助我寻找正确的方向。感谢冯洋学长，从本科毕设到研究生毕设都给我了耐心指导和细心帮助。同时特别感谢刘嘉老师、房春荣老师一直以来的关心和专业建议，我从他们身上学习到很多。还要感谢实验室的学长学姐们提供的帮助，他们的经验让我少走了许多弯路。

幸得众挚友，朝夕相处，相互扶持，学习玩乐，生活何其精彩!

感恩父母，悉心教诲，一路鼎力支持，在我遇到困难时做坚强的后盾。

最后感谢自己一直以来的坚持，让诸多努力结为果实。

至此，求学生涯画上圆满的句号。

参考文献

[1] 周强. 最高人民法院工作报告 [J], .

[2] PIPINO L L, LEE Y W, WANG R Y. Data quality assessment[J]. Communications of the ACM, 2002, 45(4): 211 – 218.

[3] CAI L, ZHU Y. The challenges of data quality and data quality assessment in the big data era[J]. Data science journal, 2015, 14.

[4] 曹建军, 刁兴春, 汪挺, et al. 数据质量控制研究中若干基本问题 [J]. 微计算机信息, 2010, 26(9): 12 – 14.

[5] SAATY T L. Decision making—the analytic hierarchy and network processes (AHP/ANP)[J]. Journal of systems science and systems engineering, 2004, 13(1): 1 – 35.

[6] FENG S, XU L D. Decision support for fuzzy comprehensive evaluation of urban development[J]. Fuzzy Sets and Systems, 1999, 105(1): 1 – 12.

[7] 李德毅, 刘常昱. 论正态云模型的普适性 [D]. [S.l.]: [s.n.], 2004.

[8] 廉昊. 面向裁判文书的大数据质量检测平台的设计与实现 [D]. [S.l.]: 南京大学, 2019.

[9] 高杰. 智慧云法院司法数据的综合治理研究 [J]. 法制与社会, 2019, 18(2): 141 – 142.

[10] 高学强. 人工智能时代的中国司法 [J]. 浙江大学学报 (人文社会科学版), 2019(4).

[11] 左卫民, 王婵媛. 基于裁判文书网的大数据法律研究: 反思与前瞻 [J]. 收藏, 2020, 2.

[12] 刘鹏. 人工智能辅助裁判理论思考与路径选择——以自然语言处理为例 [J]. 收藏, 2019, 1.

[13] 刘品新. 大数据司法的学术观察 [J]. 人民检察, 2017(23): 29–31.

[14] JIANG X, YE H, LUO Z, et al. Interpretable rationale augmented charge prediction system[C] // Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations. 2018: 146–151.

[15] YAN G, LI Y, SHEN S, et al. Law Article Prediction Based on Deep Learning[C] // 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C). 2019: 281–284.

[16] HE T, LIAN H, QIN Z, et al. Word embedding based document similarity for the inferring of penalty[C] // International Conference on Web Information Systems and Applications. 2018: 240–251.

[17] DUAN X, WANG B, WANG Z, et al. Cjrc: A reliable human-annotated benchmark dataset for chinese judicial reading comprehension[C] // China National Conference on Chinese Computational Linguistics. 2019: 439–451.

[18] 林学飞. 浅析最高人民法院公报刑事案例的裁判摘要 [J]. 法治研究, 2011(5): 99–103.

[19] FELIX C, PANDEY A V, BERTINI E. TextTile: An interactive visualization tool for seamless exploratory analysis of structured data and unstructured text[J]. IEEE transactions on visualization and computer graphics, 2016, 23(1): 161–170.

[20] ALIZAMINI F G, PEDRAM M M, ALISHAHI M, et al. Data quality improvement using fuzzy association rules[C] // 2010 International Conference on Electronics and Information Engineering: Vol 1. 2010: V1–468.

[21] BALLOU D P, PAZER H L. Modeling data and process quality in multi-input, multi-output information systems[J]. Management science, 1985, 31(2): 150–162.

[22] FIRTH C P, WANG R Y. Data quality systems evaluation and implementation[J]. cambridge market intelligence ltd, 1996.

[23] KRIEBEL C H, OTHERS. Evaluating the quality of information systems[J]. design and implementation of computer based information systems, 1979 : 29 – 43.

[24] WAND Y, WANG R Y. Anchoring data quality dimensions in ontological foundations[J]. Communications of the ACM, 1996, 39(11) : 86 – 95.

[25] BATINI C, PALMONARI M, VISCUSI G. The Many Faces of Information and their Impact on Information Quality.[C] // ICIQ. 2012 : 212 – 228.

[26] BATINI C, SCANNAPIECO M, OTHERS. Data and information quality[J]. Cham, Switzerland: Springer International Publishing. Google Scholar, 2016, 43.

[27] FLESCH R. A new readability yardstick.[J]. Journal of applied psychology, 1948, 32(3) : 221.

[28] BRODA B, NITON B, GRUSZCZYNSKI W, et al. Measuring Readability of Polish Texts: Baseline Experiments.[C] // LREC : Vol 24. 2014 : 573 – 580.

[29] KIEFER C. Assessing the Quality of Unstructured Data: An Initial Overview.[C] // LWDA. 2016 : 62 – 73.

[30] 吴思远, 于东, 江新. 汉语文本可读性特征体系构建和效度验证 [J]. 世界汉语教学, 2020, 1.

[31] LAMPLE G, BALLESTEROS M, SUBRAMANIAN S, et al. Neural architectures for named entity recognition[J]. arXiv preprint arXiv:1603.01360, 2016.

[32] BRILL E. A simple rule-based part of speech tagger[R]. [S.l.] : PENNSYLVANIA UNIV PHILADELPHIA DEPT OF COMPUTER AND INFORMATION SCIENCE, 1992.

[33] RABINER L R. A tutorial on hidden Markov models and selected applications in speech recognition[J]. Proceedings of the IEEE, 1989, 77(2) : 257 – 286.

[34] TOUTANOVA K, MANNING C. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger[C] // Proceedings of the 2000 Joint SIGDAT Conference EMNLP/VLC, 63-71, 2000. 2000.

[35] MCCALLUM A, FREITAG D, PEREIRA F C. Maximum entropy Markov models for information extraction and segmentation.[C] // Icml : Vol 17. 2000 : 591 – 598.

[36] LAFFERTY J, MCCALLUM A, PEREIRA F C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data[J], 2001.

[37] 李华栋, 贾真, 尹红风, et al. 基于规则的汉语兼类词标注方法 [J]. 计算机应用, 2014, 34(8) : 2197 – 2201.

[38] PLANK B, SØGAARD A, GOLDBERG Y. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss[J]. arXiv preprint arXiv:1604.05529, 2016.

[39] PETERS M E, NEUMANN M, IYYER M, et al. Deep contextualized word representations[J]. arXiv preprint arXiv:1802.05365, 2018.

[40] MIHALCEA R, TARAU P. Textrank: Bringing order into text[C] // Proceedings of the 2004 conference on empirical methods in natural language processing. 2004 : 404 – 411.

[41] ERKAN G, RADEV D R. Lexrank: Graph-based lexical centrality as salience in text summarization[J]. Journal of artificial intelligence research, 2004, 22 : 457 – 479.

[42] PADMAKUMAR A, SARAN A. Unsupervised text summarization using sentence embeddings[J]. Dept. Comput. Sci., Univ. Texas, Austin, USA, Tech. Rep.[Online]. Available: https://www. cs. utexas. edu/~ aish/ut/NLPProject. pdf, 2016.

[43] NALLAPATI R, ZHAI F, ZHOU B. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents[C] // Proceedings of the AAAI Conference on Artificial Intelligence : Vol 31. 2017.

[44] YU L, ZHANG W, WANG J, et al. Seqgan: Sequence generative adversarial nets with policy gradient[C] // Proceedings of the AAAI conference on artificial intelligence : Vol 31. 2017.

[45] JADHAV A, RAJAN V. Extractive summarization with swap-net: Sentences and words from alternating pointer networks[C] // Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers). 2018 : 142 – 151.

[46] CARBONELL J, GOLDSTEIN J. The use of MMR, diversity-based reranking for reordering documents and producing summaries[C] // Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. 1998 : 335 – 336.

[47] LOPYREV K. Generating news headlines with recurrent neural networks[J]. arXiv preprint arXiv:1512.01712, 2015.

[48] RUSH A M, CHOPRA S, WESTON J. A neural attention model for abstractive sentence summarization[J]. arXiv preprint arXiv:1509.00685, 2015.

[49] SEE A, LIU P J, MANNING C D. Get to the point: Summarization with pointer-generator networks[J]. arXiv preprint arXiv:1704.04368, 2017.

[50] LI C, XU W, LI S, et al. Guiding Generation for Abstractive Text Summarization Based on Key Information Guide Network[C] // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). 2018.

[51] LIN C Y, HOVY E. Automatic evaluation of summaries using N-gram co-occurrence statistics[C] // Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. 2003.

[52] BATINI C, CAPPIELLO C, FRANCALANCI C, et al. Methodologies for data quality assessment and improvement[J]. ACM computing surveys (CSUR), 2009, 41(3) : 1 – 52.

[53] KRUCHTEN P B. The 4+ 1 view model of architecture[J]. IEEE software, 1995, 12(6) : 42 – 50.

[54] 美国联邦司法中心编. 法官裁判文书写作指南 [M]. [S.l.] : 中国民主法制出版社, 2016.

[55] 余秀才. 可读性是法院法律文书公开的基本前提 [J]. 朝阳法律评论, 2011, 01(9): 203 – 211.

[56] FUTRELL R, MAHOWALD K, GIBSON E. Large-scale evidence of dependency length minimization in 37 languages[J]. Proceedings of the National Academy of Sciences, 2015, 112(33): 10336 – 10341.

[57] LIU H. Dependency distance as a metric of language comprehension difficulty[J]. Journal of Cognitive Science, 2008, 9(2): 159 – 191.

[58] LIU H, HUDSON R, FENG Z. Using a Chinese treebank to measure dependency distance[J]. Corpus Linguistics and Linguistic Theory, 2009, 5(2): 161 – 174.

[59] SU J. SPACES: “抽取-生成”式长文本摘要 [R/OL]. 2021. https://github.com/bojone/SPACES.

简历与科研成果

基本信息

张朱佩田，女，汉族，1996 年 9 月出生，福建漳州人。

教育背景

2019 年 9 月 — 2021 年 6 月 南京大学软件工程 硕士

2015 年 9 月 — 2019 年 6 月 大连海事大学网络工程 本科

攻读工程硕士学位期间完成的学术成果

1. Guo, Z., Liu, J., He, T., Li, Z., & Zhangzhu, P. (2020, July). TauJud: test augmentation of machine learning in judicial documents. In Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis (pp. 549-552).