南京大学

研 究 生 毕 业 论 文

（申请工程硕士学位）

论 文 题 目	面向裁判文书的大数据质量检测平台的设计与实现
作 者 姓 名	廉昊
学科、专业名称	工程硕士(软件工程方向)
研 究 方 向	软件工程
指 导 教 师	刘嘉 副教授 何铁科 讲师

2019 年 5 月 1 日

学　　号：**MF1732079**

论文答辩日期：**2019 年 5 月 8 日**

指 导 教 师：　　（签字）

The Design and Implementation of Big Data Quality Inspection Platform for Judgment Documents

By

Hao Lian

Supervised by

Associate Professor **Jia Liu** Lecturer **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 2019

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：面向裁判文书的大数据质量检测平台的设计与实现

工程硕士(软件工程方向) 专业 2017 级硕士生姓名： 廉昊

指导教师（姓名、职称）： 刘嘉 副教授 何铁科 讲师

摘 要

在我国智慧法院的建设背景下，可由计算机存储并处理的司法数据量快速增长，人们认识到司法数据中蕴含着巨大社会价值和业务价值。裁判文书作为审判执行流程中的关键数据，不仅整合了司法业务流程中的案件信息，更为司法案例检索、类案推荐、罚金预测等新型司法业务提供了数据基础，其数据质量决定应用效果，只有数据质量达标才能充分发挥数据价值。

法院裁判文书数据以 xml 格式存储，案情和审判信息用自然语言描述。可法院目前的文书数据质量检测方法仅校验内容合规性，缺乏对文本上下文的语义分析，没有从信息层面检测数据质量。鉴于此，本文提出了裁判文书质量检测体系，分为文书结构化内容质量和文书非结构化语义质量。文书内容质量指标结合客观信息论和粗糙集的理论知识，构建了信息层面的九个维度及其度量指标，包括适配性、广阔性、细致性、遍及性、延迟性、持续性、包容性、丰富性和真实性；文书语义质量指标采用自然语言处理方法，对案情描述进行依存句法分析和语义角色标注，构建了八个文书语义特征，提出了语义贡献度模型度量语义质量。

针对文书数据量庞大的问题，本文使用 Hadoop 大数据生态组件设计并实现了裁判文书质量检测的平台。平台具有数据交互、文书解析、质量检测和访问权限管理四个模块，可提供大数据环境下的文书分布式存储和数据质量检测服务。

本文提出的质量检测体系更全面地度量了裁判文书质量，开发的大数据平台实现了文书数据量不断增长下的质量检测服务，已作为方案提交至最高人民法院信息化服务中心。

关键词： 裁判文书，数据质量，自然语言处理，大数据，Hadoop

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 Big Data Quality Inspection Platform for Judgment Documents

SPECIALIZATION: Software Engineering

POSTGRADUATE: Hao Lian

MENTOR: Associate Professor Jia Liu Lecturer Tieke He

Abstract

Under the background of China's Wisdom Court, the amount of judicial data that can be stored and processed by computers has grown rapidly. People have recognized that judicial data contains enormous social and business value. As the key data in the trial and execution process, the judgment document not only integrates the case information in the judicial business process, but also provides a data foundation for new judicial services such as judicial case retrieval, case recommendation, and penalty prediction. The data quality determines the upper limit of the effect. Only when the data quality is up to standard can the data value be fully utilized.

The court judgment document data is stored in XML format, and the cases and trial information are described in Chinese. The court's current document data quality detection methods only verify content compliance, which lacks semantic analysis of context, and doesn't detect data quality from information level. In view of that, this thesis proposed a quality inspection system for judgement documents, which is divided into the structured content quality and the unstructured semantic quality. The content quality metrics consist of nine dimensions, which combine objective information theory and rough set theory, including suitability, broadness, granularity, coverage, delay, persistence, inclusiveness, richness and authenticity; The semantic quality adopts the natural language processing methods, the case description is analyzed by the dependency syntax analysis and the semantic role labeling. Eight semantic features are constructed, and the semantic contribution model is proposed to measure the semantic quality.

In view of the huge amount of judgement documents, this thesis uses Hadoop component to design and implement the platform for quality inspection of judgment documents. The platform has four modules: data interaction, documents analysis, quality inspection and privilege management, which can provide distributed storage and data quality inspection services in big data environment.

The quality inspection system proposed in this thesis comprehensively measures the quality of the judgment documents. The big data platform has realized the quality inspection service under the continuous growth of data. Results from the thesis have been submitted to the Supreme People's Information Service Center as a proposal.

Keywords: Judgement Documents, Data Quality, Natural Language Processing, Big Data, Hadoop

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 ····· 4

1.3.1　裁判文书质量检测体系 ····· 4

1.3.2　大数据质量检测平台 ····· 5

1.4　本文组织结构 ····· 5

第二章　技术综述 ····· **7**

2.1　数据质量 ····· 7

2.1.1　数据质量的概念及历史沿革 ····· 7

2.1.2　我国司法裁判文书数据质量 ····· 10

2.2　自然语言处理技术 ····· 12

2.2.1　分词及词性标注 ····· 12

2.2.2　依存句法分析 ····· 13

2.2.3　语义角色标注 ····· 14

2.3　Hadoop 核心组件 ····· 16

2.3.1　HDFS、Hive ····· 16

2.3.2　HBase、Zookeeper ····· 17

2.3.3　MapReduce ····· 17

2.4　本章小结 ····· 18

第三章　裁判文书质量检测体系建设及系统设计 ····· **19**

3.1　裁判文书质检体系理论分析 ····· 19

3.1.1 裁判文书内容质量检测体系 ····· 19
3.1.2 裁判文书语义质量检测方法 ····· 19
3.2 裁判文书质检体系设计 ····· 22
3.2.1 裁判文书内容质量检测指标设计 ····· 22
3.2.2 裁判文书语义质量检测指标设计 ····· 25
3.3 系统边界 ····· 29
3.4 系统需求分析 ····· 30
3.4.1 平台数据交互与文书解析 ····· 30
3.4.2 裁判文书质量检测 ····· 32
3.4.3 平台用户权限管理 ····· 34
3.4.4 非功能性需求 ····· 35
3.5 系统总体设计 ····· 36
3.6 系统详细设计 ····· 41
3.6.1 权限管理模块设计 ····· 41
3.6.2 数据交互模块设计 ····· 43
3.6.3 文书解析模块设计 ····· 44
3.6.4 质量检测模块设计 ····· 45
3.7 数据库设计 ····· 46
3.7.1 关系型数据库设计 ····· 46
3.7.2 非关系型数据库设计 ····· 50
3.8 本章小结 ····· 52

第四章 系统实现 ····· 53
4.1 文书解析 ····· 53
4.1.1 文书结构分析 ····· 53
4.1.2 文书字段解析实现 ····· 53
4.1.3 文书语义解析实现 ····· 55
4.2 文书质量检测 ····· 57
4.2.1 内容质量检测实现 ····· 57
4.2.2 语义质量检测实现 ····· 62
4.3 Hadoop分布式集群 ····· 65

4.3.1 HDFS HA高可用配置 ····· 65
4.3.2 Hive、HBase分布式配置 ····· 66
4.4 平台数据交互 ····· 67
4.5 平台用户权限管理 ····· 69
4.6 系统测试 ····· 71
4.6.1 测试环境 ····· 71
4.6.2 功能测试 ····· 71
4.6.3 性能测试 ····· 73
4.7 本章小结 ····· 74

第五章 总结与展望 ····· 75
5.1 总结 ····· 75
5.2 进一步工作 ····· 75

参考文献 ····· 77

简历与科研成果 ····· 81

致谢 ····· 83

版权及论文原创性说明 ····· 85

表　目　录

1.1 司法数据分类表 …… 1
1.2 数据质量评估框架建设方法 …… 2
1.3 司法大数据质量与传统数据质量的区别 …… 3

2.1 数据质量维度表 …… 8
2.2 国际机构和政府部门常用的数据质量维度 …… 9
2.3 数据质量完备性指标含义 …… 9
2.4 词性标注集 …… 13
2.5 依存句法标注集 …… 14
2.6 综合评估方法对比 …… 14
2.7 语义角色标注集 …… 15

3.1 文书内容质量维度表 …… 20
3.2 文书语义质量特征表 …… 22
3.3 业务系统开发人员文书内容解析用例 …… 31
3.4 平台管理员数据交互用例描述 …… 32
3.5 文书内容质量检测用例描述 …… 33
3.6 文书语义质量检测用例描述 …… 33
3.7 添加用户用例描述 …… 34
3.8 质量指标设计权限管理用例描述 …… 35
3.9 案件表case表结构 …… 48
3.10 案件结构表doc_struct表结构 …… 48
3.11 案件系统信息表case_info表结构 …… 49
3.12 集群节点信息表server_info表结构 …… 49
3.13 文书质量度量结果表case_quality表结构 …… 49
3.14 账号表account表结构 …… 50
3.15 质量指标表dimension表结构 …… 50
3.16 文书内容信息表CaseDoc表结构 …… 51

3.17 句法及语义解析结果表Analysis表结构 ····· 52

4.1 文书可提取字段信息表 ····· 54
4.2 待检数据详情表 ····· 58
4.3 文书内容质量检测结果表 ····· 59
4.4 文书质量达标率及阈值表 ····· 61
4.5 语义角色类句法树角色特征值表 ····· 63
4.6 系统测试环境 ····· 71
4.7 文书数据交互测试用例 ····· 72
4.8 文书解析测试用例 ····· 72
4.9 平台质量检测测试用例 ····· 73
4.10 平台用户权限管理测试用例 ····· 73
4.11 平台质量检测性能测试结果 ····· 74

图　目　录

2.1　数据质量计算流程 …… 8
2.2　最高院数据质量综合评价体系 …… 11
2.3　中文语义角色标注发展图 …… 15
2.4　LTP分析流程 …… 16

3.1　句法树解析图 …… 21
3.2　语义角色树状图 …… 21
3.3　质量检测平台系统边界 …… 29
3.4　平台数据交互及内容解析用例图 …… 31
3.5　裁判文书质量检测用例图 …… 32
3.6　用户权限管理模块用例图 …… 34
3.7　系统总体设计 …… 37
3.8　文书质量检测平台逻辑视图 …… 38
3.9　文书质量检测平台开发视图 …… 39
3.10　文书质量检测平台进程视图 …… 40
3.11　平台功能模块图 …… 41
3.12　数据访问权限控制模块类图 …… 42
3.13　指标设计认证模块类图 …… 43
3.14　数据交互模块类图 …… 44
3.15　文书解析模块类图 …… 45
3.16　质量检测模块类图 …… 46
3.17　关系型文书数据库实体关系图 …… 47
3.18　用户指标设计权限实体关系图 …… 47

4.1　文书字段解析实现 …… 54
4.2　文书递归解析核心代码 …… 55
4.3　文书语义解析实现 …… 56
4.4　文书语义标注结果示例 …… 56

4.5 依存句法分析核心代码 ····· 56
4.6 语义角色标注核心代码 ····· 57
4.7 文书内容质量检测实现 ····· 57
4.8 文书质量检测运行图 ····· 58
4.9 指标频率统计直方图 ····· 59
4.10 指标概率密度图 ····· 60
4.11 延迟性和持续性散点图 ····· 60
4.12 真实性频率分布图和散点图 ····· 61
4.13 文书细致性度量核心代码 ····· 61
4.14 文书语义质量检测实现 ····· 62
4.15 文书语义质量分布 ····· 63
4.16 语义角色类内句法树角色特征计算核心代码 ····· 63
4.17 语义贡献度计算核心代码 ····· 64
4.18 HDFS高可用架构 ····· 65
4.19 core.xml ····· 66
4.20 hdfs-site.xml ····· 66
4.21 hive-site.xml ····· 66
4.22 hbase-site.xml ····· 67
4.23 平台数据交互实现 ····· 67
4.24 文书下载运行图 ····· 68
4.25 文书搜索运行图 ····· 68
4.26 文书上传核心代码 ····· 68
4.27 平台权限管理实现 ····· 69
4.28 Ranger用户配置信息 ····· 70
4.29 Hive权限管理界面 ····· 70
4.30 Ranger细粒度用户授权 ····· 71

第一章　引言

1.1　项目背景

司法数据关乎民生，随着法院信息化 3.0 建设的推进，可由计算机存储并处理的司法数据量快速增长，人们逐渐认识到司法数据蕴含着巨大的社会价值和业务价值。但是海量数据的治理及应用有许多亟待解决的问题。其中数据质量把控是数据治理的关键环节之一，也是数据业务发展的瓶颈。数据质量决定了应用效果的上限，数据质量达标才能充分发挥数据的应用价值。

司法数据可分为六类，如表 1.1所示，包括审判执行数据、司法人事数据、司法研究数据、司法政务数据、信息化管理数据及外部数据。

表 1.1: 司法数据分类表

数据类别	描述	数据类别	描述
审判执行数据	起诉材料，庭审记录等	**司法人事数据**	人事信息，法院信息等
司法研究数据	指导案例，诉讼研究等	**司法政务数据**	行政规范，信息公开等
信息化管理数据	信息系统，信息化文件等	**外部数据**	涉诉信息，机构代码等

数据量丰富且对司法业务有重要影响的数据有审判执行数据和司法政务数据。审判执行数据大多为半结构化xml裁判文书类数据；司法政务数据是非结构化的文本类数据。其中，裁判文书是司法公正的最终载体，是法官向当事人和社会提交的“答卷”，是弘扬法律公正和司法威严的“法律宣言” [1]。

目前我国法院信息化 3.0 的建设已取得许多成果，“法智罗盘”集成了众多司法业务功能，是智慧法院建设成果在业务上的具体体现。其业务有法律总库查询、当事人检索、案件适用法条、文书质检、法智检索、案例检索、案件质检和前审后续8项。其中裁判文书记录了案件审理过程和结果，是实现以上审判执行相关业务的基础。

近年来，随着司法信息化进程的深入，电子化裁判文书数量骤增。最高人民法院全力推行司法公开制度，重点建立的审判流程公开网、裁判文书网、执行信息公开网和庭审公开网四大司法公开平台，迅速发展为世界最大的司法公开网络系统 [2]。截至2019年2月，中国裁判文书网公开的裁判文书量已经达到了6375万余篇 [3]。裁判文书作为法院工作改革的重要内容，备受社会关注。但是，目前全国各级法院数量庞大、各地信息化系统差异较大、司法业务复杂、

案件类型繁多，裁判文书作为审判执行流程中的重要数据，其数据质量参次不齐，需要严格把控。司法裁判文书作为我国司法数据中记录案件信息的重要载体，是案件审理环节中记录案件详情和审判过程的重要数据，其数据质量对审判执行流程有直接影响。

1.2　国内外研究现状

近年来，基于裁判文书数据和机器学习理论的应用层出不穷，例如，司法案例检索、类案推荐 [4]、罚金预测等 [5]，但是数据质量堪忧[1]。裁判文书由法官书写，书写过程中难免出现信息遗漏和错误的情况。此外，数据质量严重影响各种算法模型的精度，即使算法再先进，也被低质量数据深深束缚。智慧法院的建设离不开应用先进的机器学习方法，但我国目前缺乏完备的文书数据质量检测体系。

同时，裁判文书具有数据量大，内容呈现出以半结构和非结构化为主的特点。为了实现应用级数据质量检测功能，需要使用具有大数据特性的平台研发手段。对审判执行流程中的裁判文书数据质量进行检测，可以为法院数据中心和相关业务系统提供精确、完整的案件数据，为案件提供标准且完备的参考信息。开发面向裁判文书的数据质量检测平台对实现全国各级人民法院司法基本服务库的构建有重要作用。

数据质量的定义为“数据满足数据消费者预期的程度，即它能够把预定的使用目的或者用途完成到什么程度” [6]。

为了全面评价数据质量，需要确定数据质量的维度。但是，由于质量的应用场景不同，维度的定义存在差异。Wand & Wang等人经过实践研究提出了质量度量的六个基础维度 [6]，经过漫长发展，Batini等人凝练出了数据质量的准确性(Accuracy)、完整性(Completeness)、一致性(Consistency)、冗余性(Redundancy)、可读性(Readability)、可访问性(Accessibility)、可信度(Trust)和可用性(Usefulness) 八个维度 [7]，每个维度有对应的检测指标。

表 1.2: 数据质量评估框架建设方法

评估类别	难易程度	使用模型	应用场景	适用范围
AHP	较简单	层次结构模型	质量指标权重确定	无限制
FCE	复杂	隶属函数	模糊性质量问题	无限制
CM	复杂	正态云模型	模糊与随机性共存的质量问题	无限制

除了基本的质量测度，需要确定更高层面的数据质量检测框架。使用较

多的评估框架建设方法有三种：层次分析法(Analytic Hierarchy Process, AHP) [8]、模糊综合评估法(Fuzzy Comprehensive Evaluation, FCE) [9]，云模型评估法(Cloud Model, CM) [10]。三种方法的对比如表 1.2 所示。

本文调研发现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。但是框架中的质量检测指标仅仅适用于结构化文本，只能对文本合规性进行检查，却无法利用文本固有的句法和语义特征进行质量评估。因此，本文使用模糊集和客观信息论，构建了文书内容质量检测方法，并引入了自然语言处理技术，提出了文书语义质量检测方法。

司法大数据下的数据质量相比于传统数据质量主要存在以下四方面的挑战：复杂的异构数据源问题、信息分散问题、海量数据以及信息的可信度问题，如表 1.3 所示。

表 1.3: 司法大数据质量与传统数据质量的区别

挑战	传统司法数据质量	司法大数据质量
复杂的异构数据源	结构化数据	半结构化和非结构化数据为主
分散的信息	数据在地理和时间上集中	在地理和时间上存在隔阂
海量数据	规模较小	规模庞大
信息的可信度	交付过程明确	交付过程不明确

1. 复杂的异构数据源：传统的司法数据质量都是以结构化数据为研究对象，但是在大数据时代，非结构化和半结构化数据不断增多，占数据总量的70%以上，其数据质量的研究是难点之一 [11]。

2. 分散的信息：数据源存在地理和时间上的隔阂，导致缺乏数据约束的统一规则和方法。

3. 海量数据：传统的司法数据存储并没有考虑到数据规模，如何计算并存储指数级增长的数据是工程上的难点之一。

4. 信息的可信度：由于采集到的许多数据来自于外部数据源，平台无法清理这些数据之间的关系，更不了解这些信息是如何被交付的，因此无法得知这些数据质量处于什么水平。

因此，面向司法裁判文书的大数据质量检测平台具有学术和工程价值：

首先，构造了裁判文书质量度量体系，分为文书结构化内容质量和文书非结构化语义质量。文书内容质量指标结合客观信息论 [12] 和粗糙集 [13]的理论知识，构建了信息层面的九个指标，包括适配性、广阔性、细致性、遍及性、延迟性、持续性、包容性、丰富性和真实性；文书语义质量指标采用自然语言

处理方法，对非结构化语言部分进行依存句法分析 [14]和语义角色标注 [15]，构建了八个文书语义特征，提出了语义贡献度模型度量语义质量。

其次，使用Hadoop生态组件设计并实现了司法裁判文书大数据质量检测平台，更有利于适应数据量激增的环境。并为结构化和非结构化数据质量度量提供了分布式存储和计算方法，提升了可用性。

最后，通过平台对裁判文书质量的保障，提高了审判执行流程中与裁判文书相关的业务数据可信度。

1.3　研究目标与研究内容

本文将完成一个面向司法裁判文书的大数据质量检测平台，包含两部分内容：一是裁判文书质量检测体系建设，二是基于该体系的大数据质量检测平台。

1.3.1　裁判文书质量检测体系

裁判文书数据质量检测体系包含文书内容质量检测指标和文书语义质量检测指标两部分。内容质量关注文书表层字段的规则部分，语义质量关注文书深层的句法及语义特征部分。

1. 基于客观信息论的文书内容质量体系

文书为案件信息的载体，是案件在客观世界中的实体映射。司法数据的分类标准之一是依据案件类型进行划分，每种案件类型由不同的案由数据构成。文书作为案件细节的载体，拥有同样的划分标准，且其内容和书写格式有明确标准，因此，文书可以被解析为结构化的xml文件，其中的数据项有严格定义和要求 [16]。本文拟对数据项的内容及组织结构进行分析，使用六元信息论模型 [12]，结合粗糙集理论进行具体实现，构造适用于文书信息质量评估的九大指标体系，度量维度有广阔性、细致性、持续性、丰富性、包容性、延迟性、遍及性、真实性和适配性。

2. 基于自然语言处理方法的文书语义质量体系

为了准确地描述案件信息及其各维度的性质，需要结合自然语言处理的相关理论，挖掘文书固有的内容特征。本文采用词性标注算法 [14]探索不同案件实体的分布规律，利用基于介词逻辑的依存句法分析算法对实体间关系进行描述，使用语义角色标注算法构建基于谓词逻辑的语义表示，挖掘文书潜在的语义信息。以此构建文书语义特征，设计文书语义贡献度算法，从而检测文书语义质量。

1.3.2　大数据质量检测平台

平台面向开发者提供质量检测接口，主要由数据源管理层、数据处理层、文书质量检测层、权限管理层和接口层组成，使用Hadoop生态组件搭建。

1. 数据源管理层

数据源管理层为平台提供分布式数据存储服务，包括分布式文件系统HDFS、分布式非关系型数据库Hbase、分布式关系型数据库Hive。本层是大数据平台的底层，为结构化业务数据、非结构化和半结构化的文书数据提供存储服务，并实现集群的分布式高可用配置。

2. 数据处理层

业务处理层构建于数据服务层之上，提供文书解析服务，包括案件字段抽取和文书自然语言解析。

3. 文书质量检测层

本层是平台核心，实现了文书数据质量检测功能。为上层组件提供质检服务，包括文书内容质量检测服务和文书语义质量检测服务。

4. 权限管理层

本层提供用户权限校验功能。用户权限分为数据访问权限和指标设计权限。拥有数据访问权限的用户可以访问限定域的文书数据和业务数据；指标库中存储可被调用的质量指标名，拥有指标设计权限的用户可将设计的指标入库。

5. 接口层

接口层提供数据交互接口，是用户与平台进行数据交互的门户。

1.4　本文组织结构

本论文共分为五章：

第一章：介绍本文的研究背景和主要研究内容。

第二章：介绍本文使用的相关技术基础，包括数据质量领域的理论基础，自然语言处理领域对中文句法和语义解析的应用基础，以及Hadoop大数据组件介绍。

第三章：本章包括司法裁判文书数据质量体系设计、司法大数据平台需求分析和详细设计。质量体系包括裁判文书内容质量指标建设和语义质量指标建设；平台详细设计包括架构设计及功能模块设计。

第四章：面向裁判文书的司法大数据质量检测平台核心功能的实现。

第五章：总结及展望。总结并提出研究中存在的问题和改进方向。

第二章　技术综述

本章介绍文书数据质量体系理论基础和数据质量检测平台框架基础，包含数据质量理论、自然语言处理技术和 Hadoop 核心组件。

2.1　数据质量

本节介绍了数据质量概念及其研究内容，以及我国裁判文书数据质量检测技术的现状。

2.1.1　数据质量的概念及历史沿革

在数据质量领域，目前被普遍认同的关于数据质量的定义主要包含以下两方面内容 [17]：

1. 数据满足数据消费者预期的程度，即它能够把预定的使用目的或者用途完成到什么程度。
2. 多大程度上表示了创建它的对象、事件和概念。

对应司法数据质量，其概念为“数据满足司法业务要求的程度”。

数据质量的发展历程经历了以下四个阶段：

萌芽阶段：研究人员发现信息系统被劣质的数据质量影响，无法正常运行，但是尚未建立统一的数据质量知识体系。各行业有独立的数据质量评价体系。

形成阶段：以MIT的全面数据质量管理(Total Data Quality Management, T-DQM)理论为代表 [18]，众多学科共同创立了数据和信息质量理论，并形成了数据质量管理的统一知识体系。

繁荣阶段：数据质量研究继续深入,相关数据质量产品已经大量出现，国际组织开始研究和制定数据质量标准，政府部门颁布数据质量法案 [19]。

大数据质量阶段：大数据本身呈现出一些新特性，如4V(Volume, Variety, Velocity, Veracity)特性，因此如何从海量、快速变化、来源丰富的大数据中提取出高质量且真实的数据成为企业处理大数据过程中亟待解决的问题 [20]，也是目前该领域的研究方向之一。

为了全面评价数据质量，需要确定数据质量的维度。数据质量维度可以通过一个通用的分类框架来表征，该框架允许我们比较不同信息类型的维度特

征。该框架基于 [21]中提出的维度分类理论，质量检测指标根据它们的相似性包含在同一类中。质量维度类的定义如表 2.1所示：

表 2.1: 数据质量维度表

维度	描述
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	一致性和内聚性是指信息与现实一致的程度。例如完整性约束，业务规则和其他形式上的规定的内容。
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、清晰度和简洁性是文本信息易于理解和实现的程度。
Accessibility	可访问性和可用性与用户基于自己的文化水平，物理状态和可用技术去访问信息的能力有关。
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	可用性与用户从使用信息中获得的可用信息量有关。

数据质量计算的一般流程，如图 2.1所示。

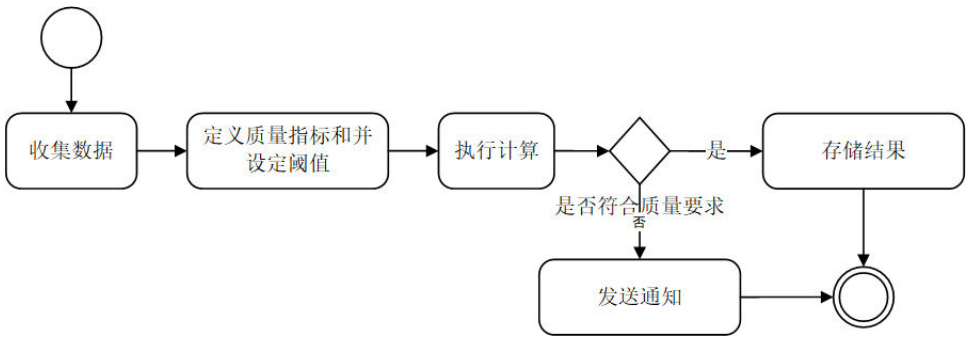

图 2.1: 数据质量计算流程

本文调研了众多国际机构和政府部门数据质量维度，为后续司法数据质量指标的构建提供参考 [22]，如表 2.2所示。

同时，本文对司法数据中可能涉及到的维度进行了调研，发现在实践中每个维度的内涵都非常丰富。

表 2.3以完备性为例，指出数据完备性的内涵。

以上的数据质量指标在很多场景下用来度量结构化的数据质量，对于非结构化和半结构化的文本数据质量，往往使用可读性进行描述。可读性对于文本

表 2.2: 国际机构和政府部门常用的数据质量维度

国际机构或者政府部门	数据质量维度
国际货币基金组织	诚信的保证、方法的健全、准确性和可靠性、适用性及可获取性
欧盟统计局	相关性、准确性、可比性、连贯性、及时性和准时、可访问性和清晰
联合国粮食及农业组织	相关性、准确性、及时性、准时性、可访问性和明确性、可比性、一致性和完备性、源数据的完备性
美联邦政府	实用性、客观性（准确、可靠、清晰、完整、无歧义）、安全性
美国商务部	可比性、准确性、适用性
美国国防部	准确性、完整性、一致性、适时性、唯一性及有效性
加拿大统计局	准确性、及时性、适用性、可访问性、衔接性、可解释性
澳大利亚国籍收支统计局	准确性、及时性、适用性、可访问性、方法科学性

表 2.3: 数据质量完备性指标含义

完备性类型	类型下的含义
数据集的完备性	元数据和参照数据的充分性 对于处理的可用性 将大小与过去的大小作比较 记录数与控制记录之比 以日期为标准 平衡经过一个流程的记录数
记录的完备性	长度
字段的完备性	不可为空字段
字段内容完备性	来自数据源的默认值 接收到的数据丢失要处理的关键字段 汇总的数额字段的比例
经过一个流程的数据集完备性	数额字段的平衡

型数据十分重要，被定义为“基于某种写作风格而被易于理解的程度” [23]。可读性研究起源于Kitson的研究，该研究表明两种英文报纸的句子长度和单词长度存在明显差异 [24]。

经典的可读性度量方法面向英文文本，为可读性而提出的大多数度量指标基于两个因素：

1. 词汇或语义特征。

2. 句子或句法复杂性。

例如，The Gunning-Fox算法是对于可读性的一种具体度量方法，其基本思想是句子越长，单词复杂程度越大，阅读文本的难度就越大 [25]。公式如下：

$$0.4 \times \left[\left(\frac{words}{sentence}\right) + 100 \times \left(\frac{complexwords}{words}\right)\right] \tag{2.1}$$

自动可读性指数(Automated Readability Index, ARI)也是一种可读性度量算法，其度量公式如下所示 [26]：

$$ARI = 4.17 \times \frac{characters}{words} + 0.5 \times \frac{complex\_words}{sentences} - 21.43 \tag{2.2}$$

与上一种度量方法不同的是，ARI算法更加关注单词或音节的复杂度。

但是，以上方法更适用于英文文本，对于中文文本，并没有明确关于复杂词或音节的定义，甚至目前中文文本语义质量度量仍没有公认的理论和方法。

随着自然语言处理领域的发展，已有工具可以对文本句法和语义进行解析。本文融合自然语言处理中的众多方法，针对上述问题提出了文书语义质量检测指标。

2.1.2　我国司法裁判文书数据质量

裁判文书是司法公正的载体，是众多办案人员在审判执行流程中逻辑思考的产物。我国司法行业标准对裁判文书格式有严格的书写要求，对表明案件审判要素的细节作了详尽要求，不同案件类型的文书要素有一定区别。例如，所有案件的裁判文书应分配并写明案号，民事二审案件中二审撤回案件应包含撤回上诉的处理事由 [27]。

按照司法行业标准书写的裁判文书可以被解析为半结构化的xml 文件，xml文件以<要素名，值>的形式存储裁判文书要素信息，无法被解析出的要素值不被列出。地方人民法院在写书写裁判文书后，以固定数量打包裁判文书并上传至最高人民法院数据汇集中心，数据中心对此批裁判文书的数据质量进行检查。如果数据质量不合要求，通知地方人民法院再次核验裁判文书。

目前我国司法裁判文书数据质量综合评价体系如图 2.2所示。

以及时率为例。在统计区间内，审判信息资源库中满足信息化数据录入要求的案件数与审判信息资源库中的全部案件数之比，称为及时率。属于正向指标，比率越大，数据的及时性越好。

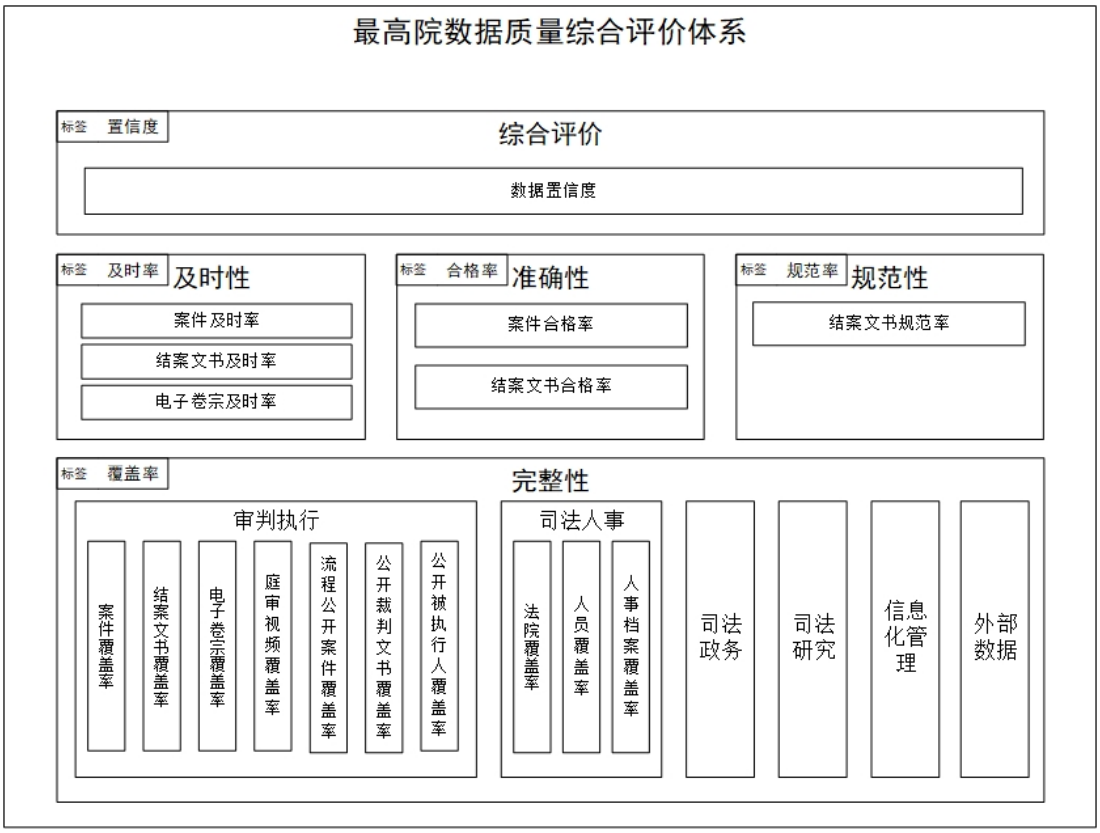

图 2.2: 最高院数据质量综合评价体系

式 2.3为及时率计算公式。

$$\Pr(x_1 \ldots x_n) = \frac{A'(x_1 \ldots x_n)}{A(x_1 \ldots x_n)} \times 100\% \tag{2.3}$$

式中，A'为审判信息库中满足信息化数据录入时间要求的案件数，A为审判信息库中的全部案件数，$A(x_1 \ldots x_n)$为审理法院、案件类型等因变量。文书的时间字段记录了审判时间。

以上内容表明，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。计算时将非结构化的裁判文书解析为半结构化的xml格式，并检查要素值的内容是否符合要求，对字段进行合规性检查。

这种检测指标无法从信息层面和文本固有的语义层面，对文本进行质量分析。为了构建更加全面的文书质量检测体系，本文采用自然语言处理技术，挖掘文本的潜在语义特征，构建语义质量检测模型，并使用信息论的知识，从信息层面度量文书质量。质量指标体系构建方法沿用层次分析法，构建基础指标或特征后，再利用其构建高层指标和特征。

2.2　自然语言处理技术

本节介绍了文书质量检测体系中用到的自然语言处理技术背景，包括分词、词性标注、依存句法分析和语义角色标注。

2.2.1　分词及词性标注

中文分词(Word Segmentation)，是中文自然语言处理时最基本的步骤 [28]。词是汉语中表达语义的基本单位，分词以词语为单位，将汉字序列切割为词序列。分词是进行信息检索、文本分类、句法解析、语义标注、情感分析等自然语言处理任务的基础。许多研究人员对中文分词进行了深入研究，产生了众多分词方法，这些分词方法大致分为基于词典匹配的方法和基于统计的方法。

基于词典的方法利用相关领域的词典库，匹配句子中的词语，但是这类方法并不能解决分词时面临的歧异问题。例如：

[原句] 林徽因在交流学术问题时遇到了梁启超

[正确分词] 林徽因/在/交流/学术/问题/时/遇到/了/梁启超

[错误分词] 林徽/因/在/交流/学术/问题/时/遇到/了/梁启超

有“林徽因”这个词的词典可以正确分词，但是没有“林徽因”的词典可能会错误分词，造成歧异。

基于统计的方法是近年来分词领域研究的重点方向，它使用专家切分好的分词语料作为训练集，通过算法训练，形成统计模型，在分词任务中对可能形成词语的组合进行概率预测，确定最优的句子解码组合。基于统计的方法使得中文分词任务在性能上有很大提升，但是其过度依赖训练集。经过多年的发展，众多领域的语料库已较为完备，这大大增加了基于统计的分词算法精度。

目前适用于中文分词任务[29]的开源工具不胜枚举，如中科院分词系统(Institute of Computing Technology, Chinese Lexical Analysis System，ICTCLAS)，哈工大语言云平台(Language Technology Platform，LTP)，清华中文词法分析工具(THU Lexical Analyzer for Chinese)等。最近发布的北大分词中文分词工具包(PKU Segmentation, PKUSeg)在公开的微博数据集上，其分词准确率达到了0.9378的高度，超越了THULAC [30]。

词性标注(Part-of-speech Tagging, POS)，是为句子中每个词语标注词性的任务 [14]，也就是要确定每个词是名词、动词、形容词或其他词性的过程。其标注流程建立在分词的基础上。

[原句] 林徽因在交流学术问题时遇到了梁启超

[词性标注] 林徽因[nh]/在[p]/交流[v]/学术[n]/问题[n]/时[nt]/遇到[v]/了[h]/梁启超[nh]

表 2.4是部分词性标注集示例。

表 2.4: 词性标注集

标签	描述	示例	标签	描述	示例
a	形容词	美丽	ni	组织名词	保险公司
b	其他名词修饰语	大型, 西式	nl	地点名词	城郊
c	结合语	和, 虽然	ns	地理名词	北京
d	副词	很	nt	时间名词	近日, 明代
e	感叹语	哎	nz	其他专有名词	诺贝尔奖
g	语素词	茨, 甥	o	拟声词	哗啦
…	…	…	…	…	…

词性是对词的一种泛化，词性在一定程度上表明了词语在当前句子中承担的角色 [31]，是句法分析和语义角色标注的必经步骤，也是本文后续语义质量指标模型构建的步骤之一。

2.2.2　依存句法分析

依存句法(Dependency Parsing, DP) 分析通过分析语言单位内成分之间的依存关系表明其句法结构 [14]。该结构以谓词为核心表征语言单位各组分的关系，通过树形结构体现句子逻辑 [32]。例如：

[原句] 委员会明天将要通过此议案

[依存句法分析] 学校[HED]/明天[SBV]/要[ADV]/评审[root]/本学期[ATT]/三好学生[POB]

整个句子的核心是“评审”，其他词语与“评审”建立起了句法关系。在句子中，“明天”的句法标签是“ADV”，“评审”的时间是“明天”。部分依存句法标签如表 2.5所示。

目前主流的依存句法分析方法有以下四类：生成式、判别式、决策式和约束型。并且短语一直是句法分析的主要研究对象。英文相对于中文更加形式化，句法结构明确，对于一个词在句中应该是单数还是复数，动词还是名词，没有太大争议，因此其句法结构较为清晰。已有许多开源工具可对中文进行句法分析，以哈工大语言云平台为代表的LTP中文自然语言处理工具包中，拥有句法分析模块，该模块使用神经网络依存句法分析算法，结合训练集全局特征和聚

表 2.5: 依存句法标注集

标记	说明	标记	说明
ADV	adverbial,default tag	MNR	Manner
BNE	benificiary	PRP	purpose or reason
CND	condition	TMP	temporal
DIR	direction	TPC	topic
DGR	Degree	CRD	coordinated arguments
EXT	extent	PRD	predicate
FRQ	frequency	PSR	possessor
LOC	locative	PSE	possessor

类特征，可以对中文句法进行有效分析，其在测试集的LAS值达到了0.8140的高度 [33]。

2.2.3　语义角色标注

语义角色标注(Semantic Role Labeling)是目前语义分析的一种主要实现方式[15]，它采用“谓语动词－角色”的结构形式，以句子谓词为中心，研究句子中各成分与谓词之间的关系，并用语义角色来描述此种的关系。每个语义角色被赋予一定的语义含义。例如：

委员会明天将要通过此议案。

[委员会Agent][明天Tmp]将要[通过V][此议案Passive]”

其中，“通过”是谓语动词，“委员会”、“此议案”和“明天”分别是其施事、受事和动作发生时间。“将要”作为与语义无关的成分被剔除。可以理解为将能表明句子含义的主干抽出。

人对于句子的理解主要通过能代表该句子含义的元素来体现，即“who did what to whom, when, where and why.”

“who, what, when, where, why”被称为论元，中文论元成分包括三类：

表 2.6: 综合评估方法对比

成分	标签
Predicate	REL
Core Arguments	ArgN
Semantic Adjuncts	ArgM-XXX

谓词或形容词是整个句子的核心，与谓词直接相关的论元用ArgN 来表示，称为核心论元；不与谓词直接相关的论元称为附加论元，用ArgM-XXX来表示，例如时间、地点、目的，程度和范围等[34]。

语义角色标注集如表 2.7所示：

表 2.7: 语义角色标注集

标记	说明	标记	说明
Arg0	施事	DIS	标记语
Arg1	受事	DGR	程度
Arg2	范围	EXT	范围
Arg3	动作开始	FRQ	频率
Arg4	动作结束	LOC	地点
Arg5	其他动词相关	MNR	方式
C-A0(LTP)	Arg0的补充	PRP	目的
C-A1(LTP)	Arg1的补充	TMP	时间
ADV	状语	TPC	主题
BNF	受益人	DIR	方向
CND	条件		

中文语义角色标注同样经历了漫长的发展历程，如图 2.3所示。

Penn Treebank(PT)是被广泛应用的英文句法分析标记库，可以通过它的句法注释可以识别句子的成分及结构，例如上面的例子。随后在其标注框架的基础上，诞生了中文的Penn Chinese Treebank(PCT) [35]。

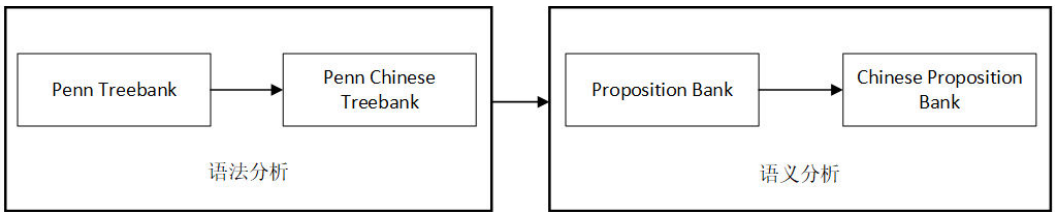

图 2.3: 中文语义角色标注发展图

虽然PT为某些成分（通常是句法附加词）提供了诸如时间和位置的语义功能标签，但它并未区分上述例子中动词的句法主语或宾语所扮演的不同角色。因此，为了进行语义层面的标注，在PT的基础上，学者们对语料进行语义标注，诞生了Proposition Bank(PB) [36]。

为了适应中文领域语义理解的发展，国内研究人员使用Proposition Bank的标注框架，在Penn Chinese Treebank基础上进行了手工中文语义角色标注，标注数据来源于新华新闻专线、Sinorama新闻杂志和香港新闻。

在国内，哈工大开发了中文自然语言处理开源工具LTP，在此基础上实现了语义角色标注功能，其使用流程如图 2.4所示：

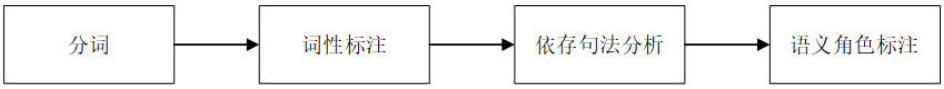

图 2.4: LTP分析流程

其语义角色标注的F1-Score达到77.15% [15]。

2.3　Hadoop 核心组件

本节介绍文书数据质量检测平台用到的Hadoop生态组件，包括HDFS、Hive、HBase、Zookeeper和MapReduce。

2.3.1　HDFS、Hive

为解决海量数据的存储问题，Google开发了分布式文件系统GFS。HDFS是GFS的开源实现，它是Hadoop的核心组件之一。HDFS 提供了在通用硬件集群中进行分布式文件存储的方法，是高容错性和高吞吐量的海量数据存储解决方案 [37]。

HDFS(Hadoop Distributed Filesystem, HDFS)以流式数据访问模式来存储超大文件，运行在由普通机器组成的集群上，是管理跨多台计算机存储的文件系统。它的基本原理是将文件切分成同样大小的数据块，存储到多台机器上，将数据切分、容错、负载均衡等功能透明化。

HDFS上的文件被划分为相同大小的block，以块作为独立的存储单位。这样做有三个原因：

1. 一个节点无法保存较大文件，需要分块存储。

2. 网络传输不稳定，如果中断可以实现小部分重传。

3. 简化存储管理，元数据无需与块一同存储，用一个单独的系统便可管理这些块的元数据。

所以block块是HDFS中最基本的存储单位。除了将文件分块，每个块文件也有副本，这是为了容错性。当一台机器宕机，可以去其他机器找文件的副本恢复里面的文件。默认是三个副本，也可以通过配置文件进行修改。

HDFS采用副本策略，将其中一个副本放在本地节点，另一个副本放到本地机架上的另一个节点，第三个副本放到其他机架上的节点。这种方式减少了机架间的写流量，从而提高了写的性能。机架故障概率小于节点故障率。在不同的机架上放置副本，防止了机架故障时数据丢失。

总之，HDFS是为了存储超大文件而设计的。它采用流式数据访问方式，特点是一次写入多次读取。其运行在普通硬件之上，即使硬件故障，也可以通过副本冗余容错机制来保证数据的高可用性。

2.3.2　HBase、Zookeeper

HBase建立在HDFS基础之上，是Google BigTable的开源版本，提供了实时读写的分布式列存储开源数据库。具有高可靠性、高性能和可扩展性。HBase是Apache的Hadoop项目的子项目。HBase中每张表的记录数可多达几亿条。每条记录可以拥有多达上百万的字段。而这样的存储能力却不需要特别的硬件系统，普通服务器集群即可胜任。HBase 的技术特点如下 [37]：

1. 大表：一个表甚至有几亿行，上百万属性列，提供海量数据存储能力，可提供高达几百亿条数据记录的存储服务。

2. 列式存储：面向列族(Column Family)的存储和控制权限管理，列族独立检索。

3. 表数据是稀疏的多维映射表：表中数据通过一个行关键字(Row Key)、一个列关键字(Column Key)以及一个时间戳进行索引和查询定位，时间戳允许数据有多个版本。

4. NoSQL数据库的典型产品：更加简单的数据模型，主要用来存储非结构化和半结构化的数据，不存在复杂的表与表之间的关系，不支持事务。

HBase提供Native Java API、HBase Shell、Thrift Gateway、Hive等多种访问方式。HBase实际使用HDFS作为底层数据存储方式，提高了数据可靠性。

ZooKeeper对HBase至关重要。ZooKeeper是HBase Master的HA(High Available, HA)解决方案。几乎所有大数据相关的开源框架，都依赖ZooKeeper实现HA，ZooKeeper 保证了至少有一个HBase Master处于运行状态。HBase客户端借助于ZooKeeper 来获得集群的Region位置信息，需要在ZooKeeper中注册Region Server。ZooKeeper实时监控每个Region Server的状态并通知给Master。

2.3.3　MapReduce

海量数据在单机上处理时，因硬件资源限制而无法胜任，而一旦将单机版

的程序扩展到集群分布式运行，将极大的增加程序的复杂度和开发难度。所以分布式并行计算编程模型MapReduce氤氲而生 [38]。

MapReduce是分布式并行计算的编程框架，是开发Hadoop相关的数据分析应用时的核心框架。MapReduce核心功能是将用户编写的业务逻辑代码和自带的组件整合成一个完整的分布式运算程序，并发运行在一个Hadoop集群上。引入MapReduce框架后，开发人员可以将绝大部分工作集中在业务逻辑的开发上，而将分布式计算中的复杂性交由框架来处理。

MapReduce由称为Map和Reduce 的两部分程序组成，实现map()和reduce()两个函数，即可实现分布式计算。这两个函数的形参是<key，value>对，表示函数输入信息。它在计算机集群上根据需求运行多个程序实例来处理子任务，再归并结果。所以开发人员只需要关注如何使用这两个函数编程实现并行计算任务，不需要处理编程中其他复杂问题。

2.4　本章小结

本章首先介绍了数据质量的定义及内涵，明确了司法大数据环境下裁判文书数据质量检测的特点和难点，然后介绍了中文句法分析和语义分析的概念和常用工具，为接下来构建中文句法和语义特征打下理论基础，最后从存储和计算的角度介绍了本文大数据平台开发的主要工具。

第三章　裁判文书质量检测体系建设及系统设计

本章首先对裁判文书质量检测体系进行了理论分析，并依此构建了具体的文书质检体系；然后明确了数据质量检测平台的系统边界，并对其进行需求分析，确定了功能模块；最后针对各功能模块进行详细设计，并给出数据库表结构，为系统实现做铺垫。

3.1　裁判文书质检体系理论分析

文书质检体系分为内容质量检测体系和语义质量检测体系，本节对此进行了理论分析，给出了体系建设方法论，为详细设计做好了铺垫。

3.1.1　裁判文书内容质量检测体系

司法数据的质量可以通过文书的信息质量进行衡量。客观信息论和自然语言处理的发展为制定文书信息质量度量的标准提供了可行的方法。

客观信息论定义信息是对客观世界和主观世界中事物及其运动状态的客观反映，物理上的各类信息系统能够且只能够获取、传输、处理并应用各种客观存在的信息 [12]。文书为案件信息的载体，是案件在客观世界中的实体映射。为了准确地描述案件信息及其各个维度的性质，本文借助六元信息论及粗糙集[13]的相关知识对文书数据质量评价标准进行了构造和量化。

裁判文书数据可以依据案件类型划分，每种案件类型包括众多案由。文书作为案件信息的载体，拥有同样的划分标准，且其书写格式以及内容有明确要求[39]。因此，文书可以被解析为结构化xml 文件，其中的数据项有严格的定义和要求。本文通过对数据项内容及组织结构进行分析，使用六元信息论的基本模型，结合粗糙集理论的具体实现，构造了适用于文书信息质量评估的九个评测指标，指标及其核心思想如表 3.1所示：

3.1.2　裁判文书语义质量检测方法

依存句法分析的标注结果无法表达更深层的语义信息，例如：

(1) John broke the window.

(2) The window broke.

表 3.1: 文书内容质量维度表

文书内容质量维度	描述
信息的适配性度量	信息价值最终取决于其对信息需求的满足程度。对于文书，信息需求的底线为案件基本信息项必须在文书中有所体现，但由于人工填写等因素，导致基本信息项缺失或错误。因此本文使用信息项的完整性定义文书信息的适配性
信息的广阔性度量	信息是对本体状态的客观反映，其价值与本体的覆盖范围直接相关。本文使用文书中的信息项覆盖程度表示其广阔性
信息的细致性度量	信息细致程度可以理解为本体能够分解成颗粒的粗细程度。文书解析为xml 文件后具有了树状结构，树的深度可以反映出该文书对案件描述的细致程度。本文使用树的深度特征，定义本体颗粒细致性
信息的持续性度量	信息的价值也与其本体的持续程度相关。本文使用单位时间内文书包含的案件实体数量定义其持续性
信息的丰富性度量	文书信息的丰富性定义为单位数量的实体数据中包含的实体类别个数。本文使用实体类别密度定义其丰富性
信息的包容性度量	使用信息熵对文书中载体容量的需求进行量化，在其他条件一定的情况下，使用载体容量需求定义其包容性
信息的延迟性度量	信息价值与其时效密切相关，文书中记录了案件发生时间和立案时间。本文使用这两者的时间差定义文书信息的延迟性
信息的遍及性度量	文书的遍及性与事件载体的分布范围相关。本文使用案件载体的分布范围定义其遍及性
信息的真实性度量	文书作为案件的映射，其真实性为对案件详情的还原程度，其可使用前文所述指标加权和进行定义

句法分析将window表示为动词在第一句中的直接对象（动宾关系），而在第二句中表示其主题（主谓关系），但并不表示它在两种情况下都扮演着相同的基础语义角色。在语义角色分析中，window都作为动作的受事（Arg1）。

再如：

(1) The sergeant played taps.

(2) Taps played quietly in the background.

句法分析在两个句子中会将taps识别为宾语和主语，但是不会将其标注为相同的语义角色，在语义角色标注中，taps均为动作的受事（Arg1）。即句法分析关注其结构，而语义角色标注关注其含义。

在结构上，一个句子对应的句法树有$n, n \in N^+$层，而其语义树有$n, n \in \{1, 2, 3\}$层，如图 3.1 所示。

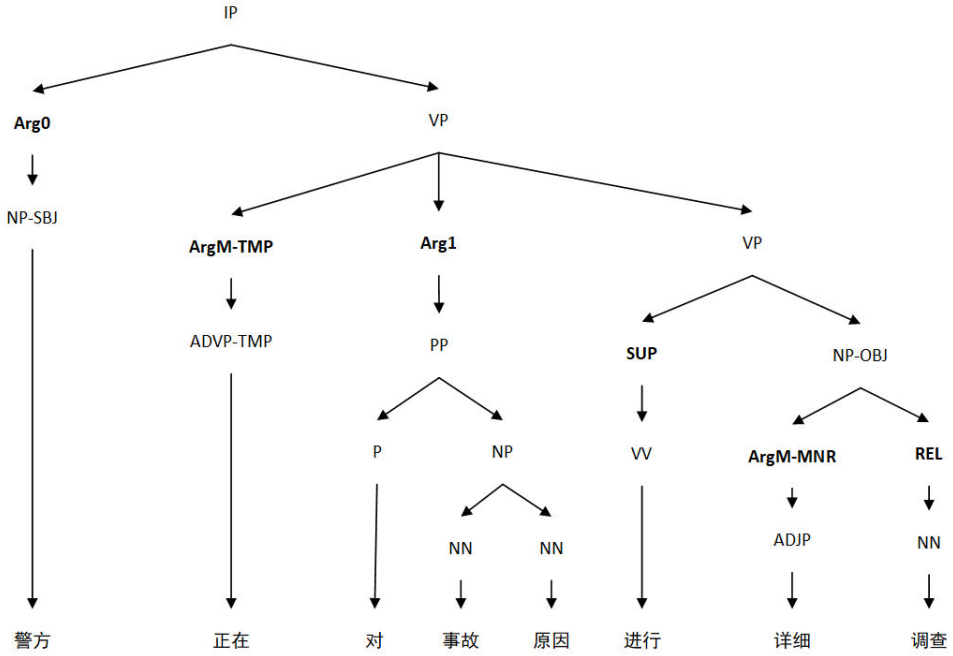

图 3.1: 句法树解析图

语义角色标注任务是在句法树基础上进行的，因此其标注任务也是在句法树节点上进行。可以看到句法树层数较多，而且比较复杂，将其中的语义角色抽离出来后如图 3.2所示：

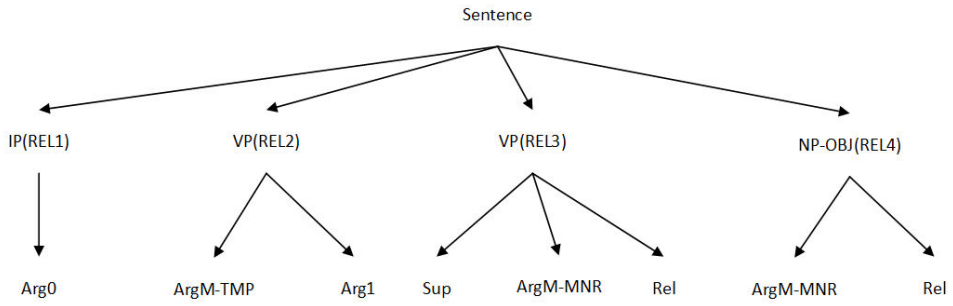

图 3.2: 语义角色树状图

因此，句法依存更重视非实词（如介词）在句子结构分析中的作用，而语义依存更倾向在具有直接语义关联的实词之间建立直接依存弧，非实词作为辅助标记存在。

两者依存弧上标记的语义关系完全不同，语义依存关系是由论元关系引申归纳而来，可以用于回答问题，如我在哪里喝汤，我在用什么喝汤，谁在喝汤，我在喝什么。但是依存句法分析却没有这个能力。

因此，为了更全面的构建中文语言特征，本文综合使用两种方法的优点，构建了8个文书语义质量特征，如表 3.2所示。

表 3.2: 文书语义质量特征表

语义特征	描述
谓词句法树角色特征	该指标关注语义谓词在句法树中充当的角色。
谓词词性特征	该指标关注语义谓词在句法树中的词性特征。
谓词数目特征	语义谓词本身是一个有限集合，该指标关注语义谓词本身的数量特征。
语义角色特征	该指标关注在裁判文书中语义角色所占的比例。
语义角色范围内的句法树角色特征	该指标关注语义角色范围内词的句法树角色比例特征。
语义角色范围的词性特征	该指标关注所有语义角色范围内词的句法树角色比例特征。
语义角色类内句法树角色特征	该指标关注每种语义角色范围内的句法角色比例特征。
语义角色类内的词性特征	该指标关注每种语义角色范围内的词性角色比例特征。

利用上述特征，本文构建了面向句子的语义贡献度计算方法：首先构建了词语的句法角色贡献度，然后用其构建短语的语义贡献度，最后定义了句子的语义贡献度。句子的平均语义贡献度在一定程度上反映了文书的语义质量。

3.2　裁判文书质检体系设计

3.2.1　裁判文书内容质量检测指标设计

设U为裁判文书全量信息集，R为文书类别划分的关系集，则在当前司法文书语料的知识库为$K = (U, R)$。

1. 信息的适配性度量：

若$P \subseteq R$，则$\cap P$为一个等价关系，称为P 上的不可区分关系，记为$ind(P)$。这样$U/ind(P)$ 代表在当前知识库K中关于裁判文书全量信息集U的P 基本概念，反映在文书中即为该案件类型下所必须具备的信息项。设X为一篇文书包含的信息集，则该篇文书的适配度定义为：

$$\text{suitability}(X) = \frac{|X/ind(P)|}{|U/ind(P)|} \tag{3.1}$$

其中，$|X/ind(P)|$代表文书X中包含的必填信息项个数，$|U/ind(P)|$代表在该类案件下所有的必填信息项个数。

2. 信息的广阔性度量：

设X为某篇文书包含的信息集，则$X \subseteq U$，其广阔度为$broadness(X)$，在X上关于文书的测度定义为：

$$\text{broadness}(X) = \frac{1}{|R|}\left|\frac{ind(P)}{U}\right|\sum_{i=1}^{n}\left\|\underline{R_i}X\right| - \left|\overline{R_i}X\right\| \tag{3.2}$$

其中$|R_n|$代表关系集的势，$\left|\underline{R_i}X\right|$ 和$\left|\overline{R_i}X\right|$代表该文书在第$i$类案由下必包含的信息项个数以及可能包含的信息项个数，表示$\left\|\underline{R_i}X\right| - \left|\overline{R_i}X\right\|$在关系$R_i$ 下处于边界域的信息项个数，$|ind(P)/U|$指该案由下必须包含的基本数据项个数的倒数。$broadness(X)$越大，代表该文书涉及到的概念域越广阔。

3. 信息的细致性度量：

设X为某篇文书包含的信息集，r为文书段落分解的规则集，则X/r为规则r下的知识集，其中的元素代表了分段后文书在该段落下的信息。在X上定义指标集$\Lambda = \cup\{layer|layer = 1, 2, 3, ..., t\}$，$\Lambda$代表文书解析为xml文件后每个段落具有的最大层数。最后细致度被定义为：

$$\text{granularity} = \frac{1}{m}\sum_{i=1}^{m}\Lambda_i \tag{3.3}$$

4. 信息的遍及性度量：

设X为某篇文书包含的信息集，o为词类的规则集，我们取其中涉及到文书实体的信息构成实体信息集X_s，显然，$X_s \subseteq X$，$X_s \in X/o$。则文书实体信息的遍及度定义为：

$$\text{coverage}(X) = \frac{|X_s|}{|X|} \tag{3.4}$$

文书中实体个数比例反映了文书要素的遍及度，遍及度越高，在一定程度上文书冗余越少。

5. 信息的持续性度量：

设X为某篇文书包含的信息集，o为词类的规则集，我们取其中涉及到案件实体的信息构成实体信息集X_s，显然，$X_s \subseteq X$，$X_s \in X/o$。T 为X中涉及到的时间集合，则信息的持续性定义为：

$$\text{persistence}(X) = \log(\frac{|X_s|}{|T|} + 1) \tag{3.5}$$

其中$|T|$代表案件的时间跨度，单位为“天”。持续性数值越高，代表在案发时间内事件密度越大。

6、信息的包容性度量：

设X为某篇文书包含的信息集，X_s为X中的实体集，q为实体类词的规则集，$a = |X/q|$，则X的包容性度量为：

$$\text{inclusiveness}(X) = \frac{1}{a}\sum_{i=1}^{a}\frac{|X_{sa}|}{|X_s|}\log(\frac{|X_{sa}|}{|X_s|}) \tag{3.6}$$

其中$|X_{sa}|$代表a类实体的个数。

$volume(X)$值越大，表达单位信息涉及到更多的事件载体，代表文书中实体出现的概率分布越均匀。

7、信息的延迟性度量：

设X为某篇文书包含的信息集，其中有立案日期和案件发生日期两个数据项，分别标记为T_end 和T_start，该文书的延迟性度量为：

$$\text{delay}(X) = \frac{1}{T_{end} - T_{start}} \tag{3.7}$$

$delay(X)$刻画了司法机关对案件的反应速度。反应越快，案件原始信息保存越完整。

8、信息的丰富性度量：

设X为某篇文书包含的信息集，其中的实体集记为X_s，实体消重后得到实体列表$X_{s-unique}$，则文书的丰富度为：

$$\text{richness}(X) = \frac{\left|X_{s-unique}\right|}{|X_s|} \tag{3.8}$$

其代表实体种类与文书中实体出现次数的比例，可以反映文书中涉及到的实体类别相对于整篇文书的丰富度，丰富度越高，代表该裁判文书可供参考的实体越多。

9、信息的真实性度量：

案例信息作为案件本身与文书之间的映射，本无真假可言，但文书所表达出的信息对于案件的还原程度是有限的，因此，我们可以综合文书的其他数据质量指标的检测结果，来对文书的还原能力作出定量评估。

$$\begin{aligned}\text{Authenticity}(X) = \ & \theta_1 \times \text{broadness}(X)+\theta_2 \times \text{granularity}(X)+\theta_3 \times \text{persistence}(X) \\ & +\theta_4 \times \text{inclusiveness}(X)+\theta_5 \times \text{delay}(X)+\theta_6 \times \text{coverage}(X) \\ & +\theta_7 \times \text{suitability}(X)+\theta_8 \times \text{richness}(X)\end{aligned} \tag{3.9}$$

3.2.2　裁判文书语义质量检测指标设计

为了能更好地表达当前裁判文书中的语义信息，下列指标将词性标注，依存句法分析和语义角色标注的三项结果相融合，给出了裁判文书在语义上的8个特征指标。

1. 谓词句法树角色特征

该特征关注语义谓词在句法树中充当的角色。N^{pre}为裁判文书中的谓词(predicate)数目，$N^{pre}_{i_{syn}}$代表在谓词中第i类句法角色(syntax)数目，$P^{pre}_{i_{syn}}$ 为第i类句法角色在裁判文书谓词中所占的比例。

$$P^{pre}_{i_{syn}} = \frac{N_{i_{syn}}}{N^{pre}} \tag{3.10}$$

例如：

(1) 他**叫**汤姆去拿外衣。

(2) 我们即将以昂扬的斗志**迎来**新的一年。

结果：叫：1/3　　拿：1/3　　迎来：1/3

2. 谓词词性特征

该特征关注语义谓词词性。N^{pre}为裁判文书中的谓词(predicate)数目，$N^{pre}_{i_{pos}}$代表在谓词中第i类词性角色(postag)数目，$P^{pre}_{i_{pos}}$ 为第i 类词性角色在裁判文书谓词中所占的比例。

$$P^{pre}_{i_{pos}} = \frac{N^{pre}_{i_{pos}}}{N^{pre}} \tag{3.11}$$

例如：

(1) 他**叫(v)**汤姆去**拿(v)**外衣。

(2) 我们即将以昂扬的斗志**迎来(v)**新的一年。

结果：v: 3/3 = 1

3. 谓词数目特征语义谓词本身是一个有限集合，该指标关注语义谓词本身。N^{pre}为裁判文书中的谓词(predicate)数目，$N_{i_{word}}^{pre}$代表在谓词中第i类谓词数目，$P_{i_{word}}^{pre}$为第i类谓词在全文谓词中所占的比例。

$$P_{i_{word}}^{pre} = \frac{N_{i_{word}}^{pre}}{N^{pre}} \tag{3.12}$$

例如：

(1) 他**叫**汤姆去**拿**外衣。

(2) 这件衣服被**拿**来了。

结果：叫: 1/3　　拿: 2/3

4. 语义角色特征

该特征关注在裁判文书中语义角色所占的比例。N^{arg}为裁判文书中的语义角色数目，N_i^{arg}为第i类语义角色数目，P_i^{arg}为第i类语义角色占总数的比例。

$$P_i^{arg} = \frac{N_i^{arg}}{N^{arg}} \tag{3.13}$$

注：语义角色指语义角色标签，语义角色范围指该标签在句子中对应的短语，“范围”与下文提到的“类”不是一个概念。

例如：

原句：进入新世纪后，经济全球化成为非洲国家面临的一个重大而严峻的挑战。

解析后：TMP（进入[ATT]新[ATT]世纪[VOB]后[ADV]，[WP]）A0（经济[ATT]全球化[SBV]）成为[HED] A1（A0（非洲[ATT]国家[SBV]）面临[ATT]的[RAD]一个[ATT]重大[ATT] 而[LAD]严峻[COO]的[RAD] 挑战[VOB]）

结果：A0 = 2/4　　TMP = 1/4　　A1 = 1/4

5. 语义角色范围的句法树角色特征

该特征关注语义角色范围内词的句法树角色。其中$N^{arg-scope-total}$代表裁判文书中所有语义角色范围内总词数，$N_{i_{syn}}^{arg-scope-total}$代表第$i$类句法角色在语义角色范围内的数目，$P_{i_{syn}}^{arg-scope-total}$ 为裁判文书中语义角色范围内词的第i类句法树角色比例。

$$P_{i_{syn}}^{arg-scope-total} = \frac{N_{i_{syn}}^{arg-scope-total}}{N^{arg-scope-total}} \tag{3.14}$$

例如：

解析后：TMP（进入[ATT]新[ATT]世纪[VOB]后[ADV]，[WP]）A0（经济[ATT]全球化[SBV]）成为[HED] A1（A0（非洲[ATT]国家[SBV]）面临[ATT]的[RAD]一个[ATT]重大[ATT] 而[LAD]严峻[COO]的[RAD] 挑战[VOB]）

结果：ATT = (7 + 1)/[(18-1) + (3-1)]　　RAD = 2/19

6. 语义角色范围的词性特征

该特征关注所有语义角色范围内词的句法树角色。其中$N^{arg-scope-total}$代表裁判文书中所有语义角色范围内总词数，$N_{i_{syn}}^{arg-scope-total}$代表第$i$类句法角色在语义角色范围内的数目，$P_{i_{syn}}^{arg-scope-total}$ 为裁判文书中语义角色范围内词的第i类句法树角色比例。

$$P_{i_{pos}}^{arg-scope-total} = \frac{N_{i_{pos}}^{arg-scope-total}}{N^{arg-scope-total}} \tag{3.15}$$

例如：

解析后：TMP（进入v[ATT]新a[ATT]世纪n[VOB]后nd[ADV]，wp[WP]）A0（经济n[ATT]全球化v[SBV]）成为v[HED] A1 （A0（非洲ns[ATT]国家n[SBV]）面临v[ATT]的u[RAD]一个m[ATT]重大a[ATT]而c[LAD]严峻a[COO]的u[RAD]挑战v[VOB]）

结果：n = (3 + 1)/[(18-1) + (3-1)]　　v = [(5-1) + (1-1)]/19

7. 语义角色类内的句法树角色特征

该特征关注每种语义角色范围内的句法角色。其中$N^{arg-scope_j}$代表第j种语义角色范围内总词数，$N_{i_{syn}}^{arg-scope_j}$代表第$i$类句法角色在第$j$ 种语义角色范围内的数目，$P_{i_{syn}}^{arg-scope_j}$为第$i$类句法角色在第$j$种语义角色范围内所占的比例。

$$P_{i_{syn}}^{arg-scope_j} = \frac{N_{i_{syn}}^{arg-scope_j}}{N^{arg-scope_j}} \tag{3.16}$$

例如上文例句解析后：

HED : ’时间’: [[’ATT’, ’ATT’, ’VOB’, ’ADV’, ’WP’]], ’施事’: [[’ATT’, ’SBV’]], ’受事’: [[’ATT’, ’SBV’, ’ATT’, ’RAD’, ’ATT’, ’ATT’, ’LAD’, ’COO’, ’RAD’, ’VOB’]]

ATT : ’施事’: [[’ATT’, ’SBV’]]

结果：

施事：ATT = 2/4　　SBV = 2/4

时间：ATT = 2/5　　VOB = 1/5　　ADV = 1/5　　WP = 1/5

8. 语义角色类内的词性特征

该特征关注每种语义角色范围内的词性角色。其中$N^{arg-scope_j}$代表第j种语义角色范围内总词数，$N_{i_{pos}}^{arg-scope_j}$代表第$i$类词性角色在第$j$ 种语义角色范围内的数目，$P_{i_{pos}}^{arg-scope_j}$为第$i$类词性角色在第$j$种语义角色范围内所占的比例。

$$P_{i_{pos}}^{arg-scope_j} = \frac{N_{i_{pos}}^{arg-scope_j}}{N^{arg-scope_j}} \tag{3.17}$$

例如上文例句解析后：

HED : ’时间’: [[’ATT’, ’ATT’, ’VOB’, ’ADV’, ’WP’]], ’施事’: [[’ATT’, ’SBV’]], ’受事’: [[’ATT’, ’SBV’, ’ATT’, ’RAD’, ’ATT’, ’ATT’, ’LAD’, ’COO’, ’RAD’, ’VOB’]]

ATT : ’施事’: [[’ATT’, ’SBV’]]

结果：

施事：n = 2/4　　ns = 1/4　　v = 1/4

时间：v = 1/5

根据以上特征，本文构建了裁判文书的平均语义贡献度，以此度量本文语义信息质量。

首先定义相同句法角色在句中的贡献度，一个词有两方面信息，词法信息和句法信息。$P_i^{arc-scope}$代表具有相同句法角色的词在句中的平均贡献度，$P_{i_{syn}}^{arg-scope}$ 代表这个词的语义角色类内的句法角色比例，n为包含第i 种句法标签的句子数目。

$$P_i^{arc-scope} = \frac{\sum_{i=1}^{n} 1/(P_{i_{syn}}^{arg-scope} \times P_{i_{pos}}^{arg-scope})}{n} \tag{3.18}$$

然后定义$P_j^{arg-scope}$为第j类语义角色在句中的平均贡献度，其中m是在句中第j类语义角色范围里包含的语义角色类型数目。

$$P_j^{arg-scope} = \frac{\sum_{j=1}^{m} P_j^{arc-scope}}{m} \tag{3.19}$$

句子贡献度$con^{sentence}$使用语义角色贡献度$P_k^{arg-scope}$定义，其中t代表该句中出现的语义角色数目。

$$con^{sentence} = \frac{\sum_{k=1}^{t} P_k^{arg-scope}}{t} \tag{3.20}$$

平均语义贡献度$ave_con^{sentence}$使用句子贡献度$con^{sentence}$来定义，其中N为本文句子总数。

$$ave\_con^{sentence} = \frac{\sum_{l=1}^{N} con_l^{sentence}}{N} \tag{3.21}$$

平均语义贡献度越高，可以在一定程度上认为本文句子的语义单位更具体，表述更加紧凑，上下文表述模式更一致。

3.3　系统边界

裁判文书大数据质量检测平台是法院信息化 3.0 系统中的一部分，具体职责是以接口形式为司法业务系统提供裁判文书质量检测服务，与如图 3.3所示。

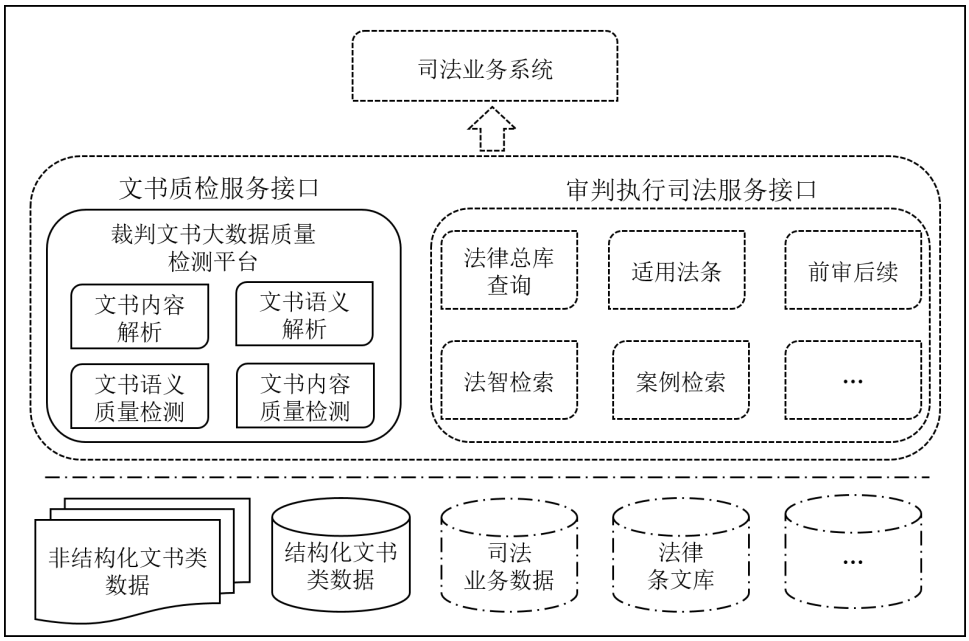

图 3.3: 质量检测平台系统边界

图中实线区域是数据质量平台的系统边界，本平台所有功能围绕裁判文书数据展开。文书数据分为非结构化文书数据和结构化文书数据。非结构化文书数据以中文文本为主，结构化文书类数据以解析后的xml数据为主。

值得一提的是，由于xml数据为半结构化数据，并且裁判文书有固定的写作规则，因此可以将文书字段以结构化方式存储至关系型数据库中。本平台使用MySQL和Hive存储关系型数据，HBase存储非关系型数据，HDFS存储原始裁判文书文件。

如图 3.3所示，除了文书类数据之外，还有司法业务数据和法律条文数据等，这些数据共同构成了司法业务系统的数据基础。文书质检平台构建并使用了其中的两个文书数据库。

在以文书质量为主题的实际应用场景中，实现“裁判文书数据质量检测平台”主要包含四项内容：

1. 实现用户与平台间接口的数据交互和分布式存储。利用Hadoop HDFS的API实现数据交互接口，在集群节点实现数据的分布式存储。NameNode节点中存储文件目录的MetaData 元数据信息，DataNode节点中存储客户端发来的Block数据块。

2. 解析裁判文书。利用裁判文书拥有xml半结构化文本的特点，提取文书原文本及关键字段信息。将信息持久化到MySQL和HBase中，为数据质量计算做铺垫。

3. 实现数据质量计算服务。使用LTP对裁判文书原文本进行依存句法分析和语义角色标注，结合文书的结构化和非结构化信息，利用MapReduce实现数据质量指标的分布式计算服务。

4. 管理用户权限。利用平台管理员的职能，使用Apache Ranger和验证机制限制用户权限，保障数据安全。

3.4　系统需求分析

基于3.3章节确立的系统边界，系统功能需求可进一步划分。本章节将对数据交互与内容解析功能、文书质量检测功能和访问权限管理功能进行详细需求分析。

3.4.1　平台数据交互与文书解析

无论是司法业务人员还是系统开发者，都需要基本的文书上传下载，以及内容解析功能。文书解析是质量检测的基础，同时，也是平台使用者获取文书内容详情的途径。

图 3.4是平台数据交互及内容解析用例图。

数据交互和文书解析功能面向业务系统开发人员和平台管理员，平台管理员具有最高权限，可访问平台所有数据。业务系统开发人员依据相应需求，赋予限定域的数据访问及接口调用权限。

表 3.3提供了业务系统开发人员调用文书内容解析功能的用例描述，表 3.4提供了平台管理员数据交互用例描述。

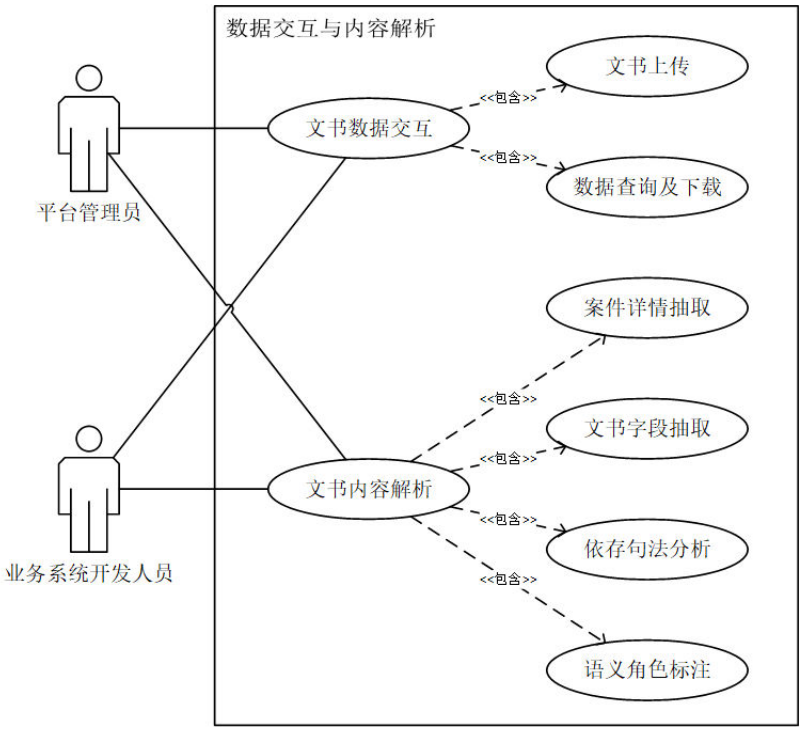

图 3.4: 平台数据交互及内容解析用例图

业务系统开发人员是平台的主要使用者，数据交互是使用平台功能的前提，文书解析是质量检测的必经步骤。

表 3.3: 业务系统开发人员文书内容解析用例

ID	UC1	**名称**	文书内容解析接口调用
描述	业务系统开发人员调用文书内容解析接口，对裁判文书详情进行解析		
参与者	业务系统开发人员		
前置条件	系统开发人员需要将被解析文书上传至平台，或者文书已存在于平台		
后置条件	平台返回文书解析结果，包括字段解析结果，句法分析结果及语义分析结果		
优先级	中		
正常流程	1. 调用测试接口，保证http连接畅通 2. 编写业务代码，调用文书上传接口或文书查询接口，获取目标文书 3. 编写业务代码，调用文书解析接口，平台抽取文书的案件详情和关键字段，并对文书进行句法分析和语义角色标注 4. 平台返回解析结果，开发人员在客户端将结果存储至本地数据库		
扩展流程	4a. 解析失败 1. 返回解析失败的错误记录，并将其记录至平台log 2. 向用户返回错误报告		

表 3.4: 平台管理员数据交互用例描述

ID	UC2	名称	文书数据交互
描述	平台管理员与平台间进行业务数据交互		
参与者	平台管理员		
前置条件	http服务连接通畅		
后置条件	平台返回交互结果标识，并将交互流程记录至log文件		
优先级	高		
正常流程	1. 调用测试接口，保证http连接畅通 2. 根据裁判文书案号字段查询裁判文书，获取原文件及解析结果；上传xml格式的裁判文书至平台 3. 平台依据操作返回对应结果，查询及获取文件返回具体文件；上传返回上传结果 4. 平台将操作记录和过程信息保存至log文件		
扩展流程	3a. 查询或上传失败 1. 返回失败原因，并将其记录至平台log文件		

3.4.2　裁判文书质量检测

此功能是平台的核心功能，它为司法人员提交的裁判文书数据提供全面的数据质量检测服务，分为文书内容质量检测和文书语义质量检测。

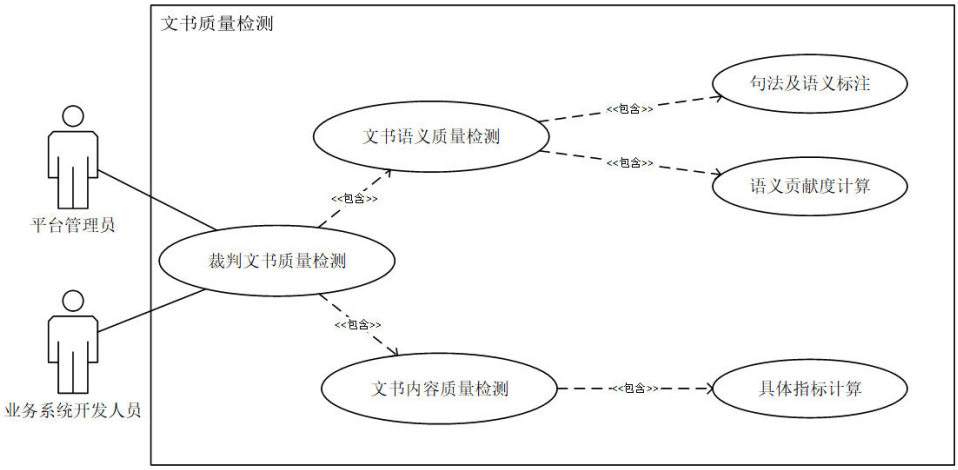

图 3.5: 裁判文书质量检测用例图

内容质量检测包含适配性、广阔性、细致性、遍及性、持续性、包容性、延迟性、丰富性、真实性。语义质量检测给出文书中每个句子的语义贡献度和语义标注集，通过语义贡献度衡量本文语义质量。文书质量度量建立在内容解

析基础上。图 3.5 是平台裁判文书质量检测用例图。本功能面向平台管理员和业务系统开发人员，表 3.5是司法业务系统开发人员调用文书质量检测接口、检测文书内容质量的用例描述。表 3.6是检测文书语义质量的用例描述。

表 3.5: 文书内容质量检测用例描述

ID	UC3	**名称**	文书内容质量检测
描述	对上传的裁判文书集合内容质量进行检测		
参与者	司法业务系统开发人员		
前置条件	裁判文书集上传成功		
后置条件	无		
优先级	中		
正常流程	1. 客户端与平台的http测试接口连接通畅 2. 编码调用文书内容质量检测接口，解析裁判文书 3. 内容解析完成后分布式计算文书各指标结果 4. 质量检测结果以JSON文件的形式返回		
扩展流程	3a. 内容解析失败 1. 提醒内容解析失败信息，并尝试1次重新解析 2. 重新解析失败将失败结果返回至用户，并在log文件中记录		
特殊需求	1. 每30篇裁判文书计算时间不超过30秒		

表 3.6: 文书语义质量检测用例描述

ID	UC4	**名称**	文书语义质量检测
描述	对上传的裁判文书集合语义质量进行检测		
参与者	司法业务系统开发人员		
前置条件	裁判文书集上传成功		
后置条件	无		
优先级	中		
正常流程	1. 客户端与平台的http测试接口连接通畅 2. 编码调用文书语义质量检测接口 3. 平台对裁判文书进行依存句法分析及语义角色标注 4. 对每一篇裁判文书计算句子的语义贡献度，导出文书语义质量 5. 质量检测结果以JSON文件的形式返回		
扩展流程	3a. 句法及语义分析失败 1. 提醒内容解析失败信息，并尝试1次重新解析 2. 重新解析失败将失败结果返回至用户，并在log文件中记录		
特殊需求	1. 每30篇裁判文书计算时间不超过60秒		

3.4.3　平台用户权限管理

为了保障数据安全，必须从平台层对平台操作权限加以管控。用户权限管理模块负责用户可访问平台内容的范围。用户权限管理可划分为两个子模块：数据访问权限控制模块、指标设计权限管理模块，如图 3.6所示。

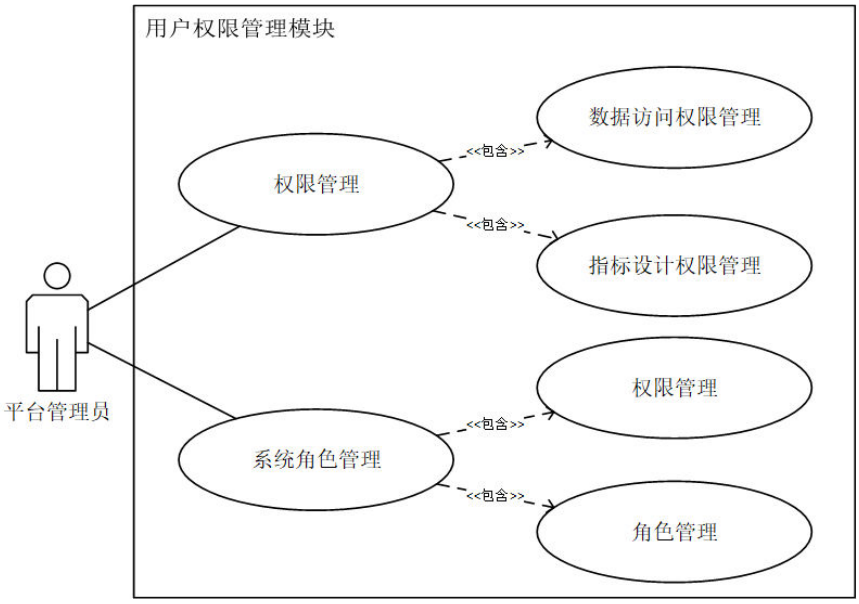

图 3.6: 用户权限管理模块用例图

表 3.7: 添加用户用例描述

ID	UC5	**名称**	平台角色管理
描述	在平台上添加相关司法业务系统的开发人员信息		
参与者	平台管理员		
前置条件	开发人员负责的业务系统中有文书质量业务需求		
后置条件	开发人员可依据个人密钥接入平台		
优先级	高		
正常流程	1. 客户端与平台的http测试接口连接通畅 2. 平台管理员确认开发人员姓名、身份证等个人信息及开发机ip 3. 利用开发人员个人信息和ip地址生成密钥并存储至可接入平台人员数据库中 4. 开发人员接入平台时验证成功		

数据访问权限控制模块对平台数据进行细粒度的权限管理。

指标设计权限管理模块控制用户对指标库的访问权限。只有存储在指标库中的指标才能被平台调用，开发者获取到指标设计权限后可将新指标入库。这

样的方式保证了指标有效性。

平台面向两类人员开放接口，一类是业务系统开发人员，另一类是平台管理员。两类人员职能不同，其对平台的操作权限也不同。对于业务系统开发人员，平台为其提供文书上传和文书质量检测接口。除了业务类接口外，还为其开放质量类设计接口，可以使开发人员不拘泥于已有检测指标，设计适合当下业务系统的质量检测指标，将指标入库；对于平台管理员，其拥有授权、维护、更改平台的一切权限，此角色的存在是为了让平台内部环境不断演进，以适应业务发展需要。

因此，平台管理员拥有平台最高权限，可以控制用户权限，并管理平台角色。图 3.6为用户权限管理模块的需求用例图。

依据模块需求，表 3.7为平台管理员添加开发人员角色的用例描述，表 3.8是平台管理员为用户赋予指标设计权限的用例描述。

表 3.8: 质量指标设计权限管理用例描述

ID	UC6	**名称**	质量指标设计权限管理
描述	在平台上为角色添加质量指标定义权限，使其能设计并实现质量指标，并将其添加到指标库中		
参与者	平台管理员		
前置条件	开发人员在平台接入人员列表中		
后置条件	无		
优先级	高		
正常流程	1. 开发人员提出有质量规则制定的业务需求，需要增加质量评定指标 2. 平台管理员为其开放质量规则接口权限，可向质量指标库添加指标，并实现质量接口		

3.4.4　非功能性需求

1. 响应性能

平台为众多司法业务应用提供服务接口，因此需要考虑并发性及每次请求的响应时间。尤其是计算服务，要及时反馈质量检测结果。

2. 安全性

系统内部存储了众多司法文书数据，这些都是基础且重要的数据，一旦遭到破坏，对业务应用也有很大影响。因此，所有外部请求必须进行权限验证和过滤。

3. 容错性

系统中的数据应考虑单点失效造成的影响，应使用冗余手段备份数据，配置高可用的分布式集群。使得任意一台服务器出现问题不会影响系统的整体可用性。

3.5　系统总体设计

面向裁判文书的大数据质量检测平台的总体结构如图 3.7所示，包括数据源管理层、数据处理层、裁判文书质量检测层、权限管理层和接口层。本节对平台进行了总体设计，确定了系统框架。

接口层主要为开发者提供三类接口：质量检测接口、质量指标设计接口和数据交互接口。质量检测接口为开发者提供文书内容质量检测服务和文书语义质量检测服务；质量指标设计接口为开发者提供灵活的质量指标设计方法，以适应环境需要；数据交互接口用于平台与开发者的数据交互。

权限管理层使用Apache Ranger作为数据权限管理框架，使用MySQL存储接口访问权限。将Ranger以插件形式集成到HDFS中，由Ranger Admin管理。平台管理员通过Ranger Web 登陆Ranger Admin，为用户配置策略，控制用户访问HDFS文件夹，数据库，表的权限。同时，用户对接口的访问权限存储在MySQL中。用户只有拥有对接口的访问权限，才能正常调用接口，Ranger为数据安全提供了保障。

质量检测服务层是平台核心，使用MapReduce提供分布式计算服务。在集群的各个节点上，依据接口调用参数执行计算：首先依据参数在HDFS中找到目标文书集合，然后在相应的DataNode上检测文书质量，最后使用MapReduce服务在NameNode返回计算结果。质量检测层度量裁判文书的内容质量和语义质量。内容质量有9个质量指标，从适配性、广阔性、细致性、遍及性、持续性、包容性、延迟性、丰富性和真实性9个维度对裁判文书内容质量作出评估；语义质量通过语义贡献度定义，综合了词性标注、依存句法分析、语义角色标注的结果，通过构建的8种语义特征进行建模。

数据处理层介于数据源管理层和质量检测服务层之间，本层的职责有三：一是抽取关键字段，将半结构化xml文书依据字段抽取出重要内容，例如案号，将其存储至集群节点中；二是对文本进行解析，递归获取所有文书内容，为内容质量计算做准备；三是对文本内容进行依存句法分析和语义角色标注，为语义质量计算做准备。本层数据处理的部分内容需要存储至数据库，即与数据源管理层交互。

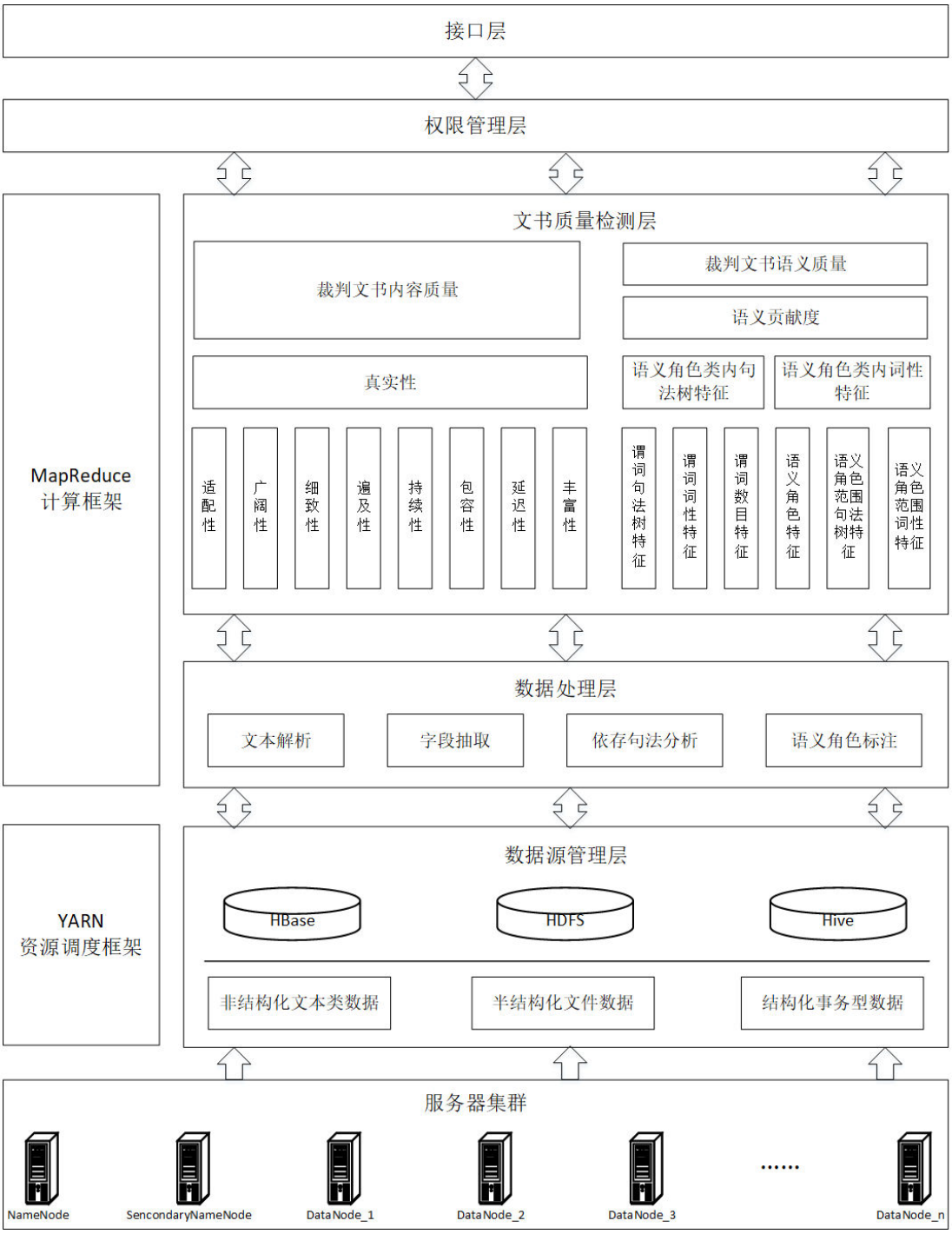

图 3.7: 系统总体设计

数据源管理层由HBase、Hive和HDFS组成，其中HDFS是HBase和Hive 实现的基础。在Hive中存储了所有结构化信息，例如案件的案号、案由和罚金等；在HBase中存储了所有非结构化信息，例如文书原文本、文本语义分析结果等。

本层向数据处理层提供元数据。

平台的所有数据存储在服务器集群中，集群分为NameNode和DataNode，NameNode负责接口管理，权限管理等，DataNode负责存储数据和执行计算。

MapReduce和YARN作为横贯平台的两大框架，MapReduce提供了分布式计算服务，YARN使得HDFS资源管理更加统一，方便平台管理员制定策略。

如图 3.8是文书质量检测平台的逻辑视图。平台核心由五个包组成。Web包提供平台与用户交互的常用接口和界面，集成了平台API组件。为获取集群信息，对节点进行监控，管理集群配置信息，需要nodeconfig包统一总管集群底层信息。文书交互和解析，字段和计算结果存储等数据相关业务由data包提供。平台功能性业务由util包负责，集成了文本语言处理、文书质量检测、权限管理等组件，其依赖data包提供的数据交互和文书解析功能。

将用户逻辑和底层逻辑依据业务逻辑统一起来的是service包，其向下提供业务信息，向上依赖交互数据，充当信息管道和组件整合角色。

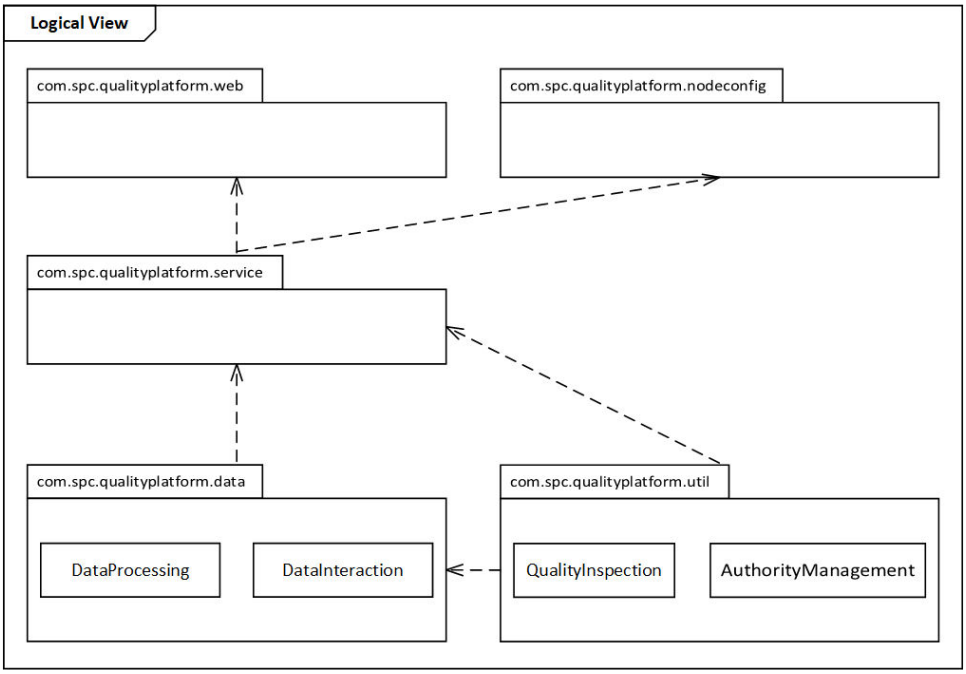

图 3.8: 文书质量检测平台逻辑视图

平台开发视图如图 3.9所示，文书质量检测平台主要由数据管理模块、文书内容解析模块、NLP处理模块、权限管理模块和质量检测模块组成。所有模块建立在Hadoop 集群基础上。所有模块使用Deployment方式部署到一个或多个node之上。Hadoop集群使用HDFS HA高可用配置方式，因此即使有node失效，集群也可以恢复该节点的数据和进程状态。

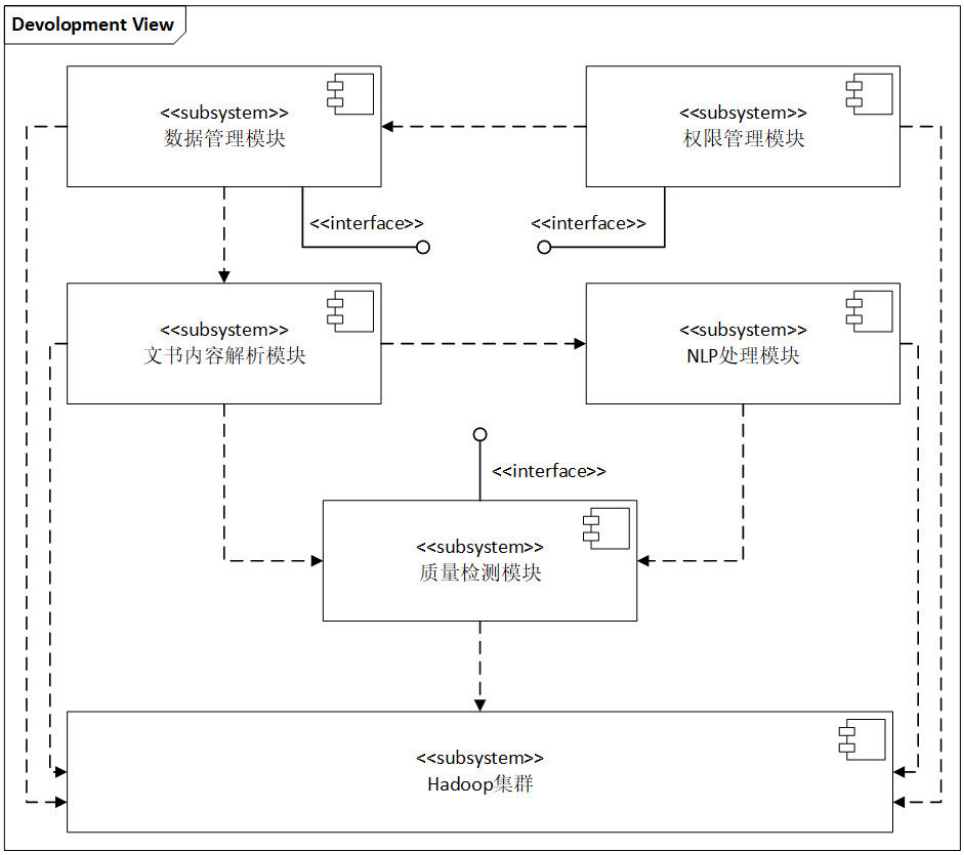

图 3.9: 文书质量检测平台开发视图

数据管理模块提供数据库操作和数据交互等功能，涉及结构化数据在关系型数据库Hive中的存储和非结构化数据在HBase中的存储，为文书解析模块提供文书数据。文书内容解析模块主要负责xml格式的文书结构和字段解析，其解析结果支撑文书内容质量计算。同时，解析出的自然语言描述部分为NLP处理模块提供数据，对案情描述进行分词、词性标注、依存句法分析和语义角色标注，其处理结果支撑文书语义质量计算。

数据管理模块、权限管理模块和质量检测模块向内提供接口，用于实现平台的业务逻辑。

图 3.10是平台的进程视图。质量检测平台的每个子系统都作为一个进程独立运行在node之上。图中展示了两个典型的交互流程。

用户质量检测交互流程主要由文书解析JudicialDocumentParse 和质量检测QualityInspect两个分片构成。首先用户将个人信息和待检数据传输至权限管理对象ac: AuthController，权限验证通过后将数据提交至文书解析对象tp: TextParse。文书解析的中间结果以异步方式递交给数据存储对象ds:

DataStorageService，负责将文书字段、案件结构信息和案件详情存储至数据库中。同时，数据提供给质量检测对象dqc: DataQualityCal进行质量检测，分为语义质量检测和内容质量检测，检测结果由ds对象存储至数据库。

质量指标设计交互是用户添加自行设计的指标至指标库的流程。只有在指标库中存在的指标，才能被调用计算。用户首先要通过权限验证，拥有权限的用户创建的dq: DataQuality对象才会被存储至指标库。

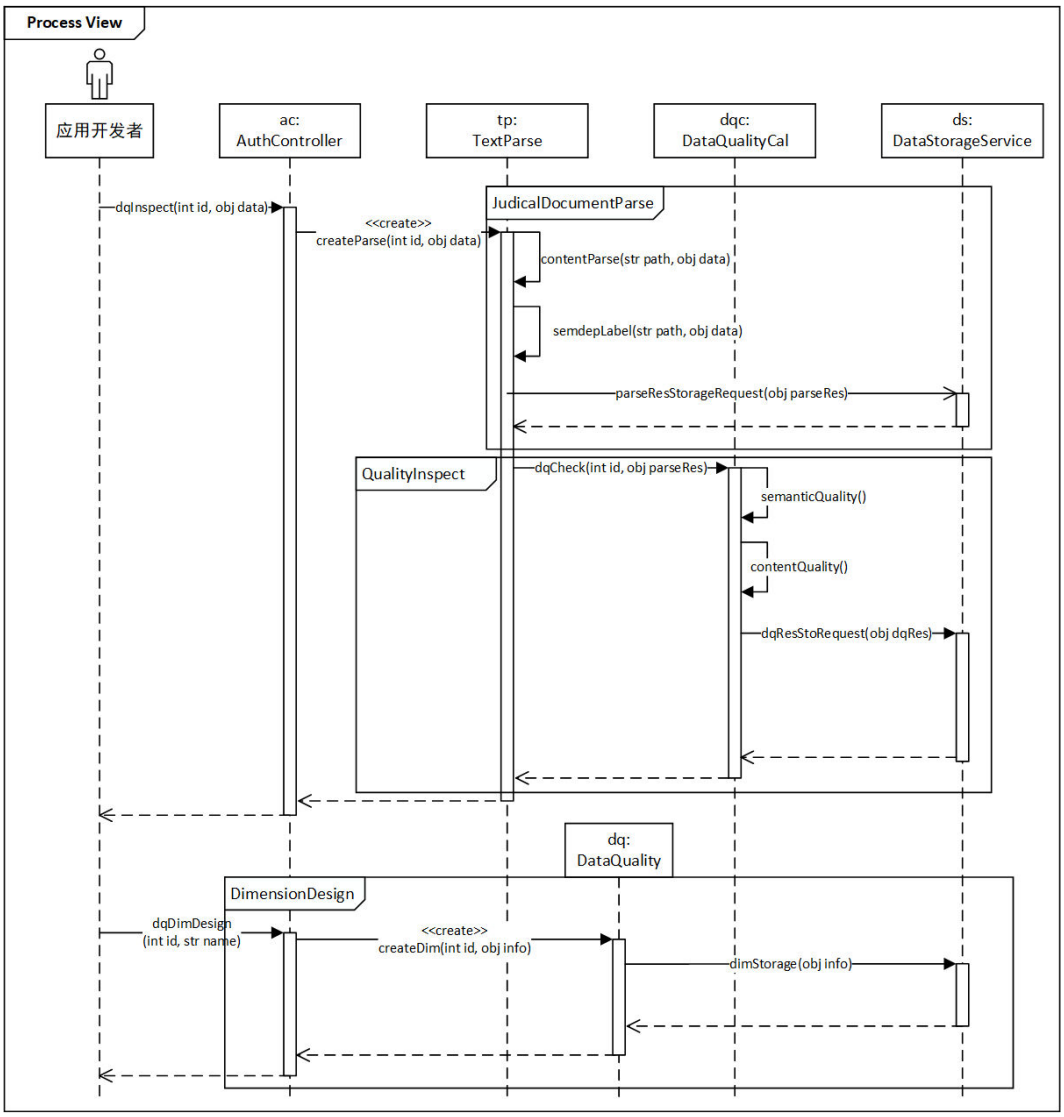

图 3.10: 文书质量检测平台进程视图

依据上述分析，平台基本功能如图 3.11所示，分为访问权限管理模块、数据交互模块、文书内容解析模块和文书质量检测模块。文书质量检测模块是平台核心。文书质检体系包含文书内容质量检测和文书语义质量检测两项内容。

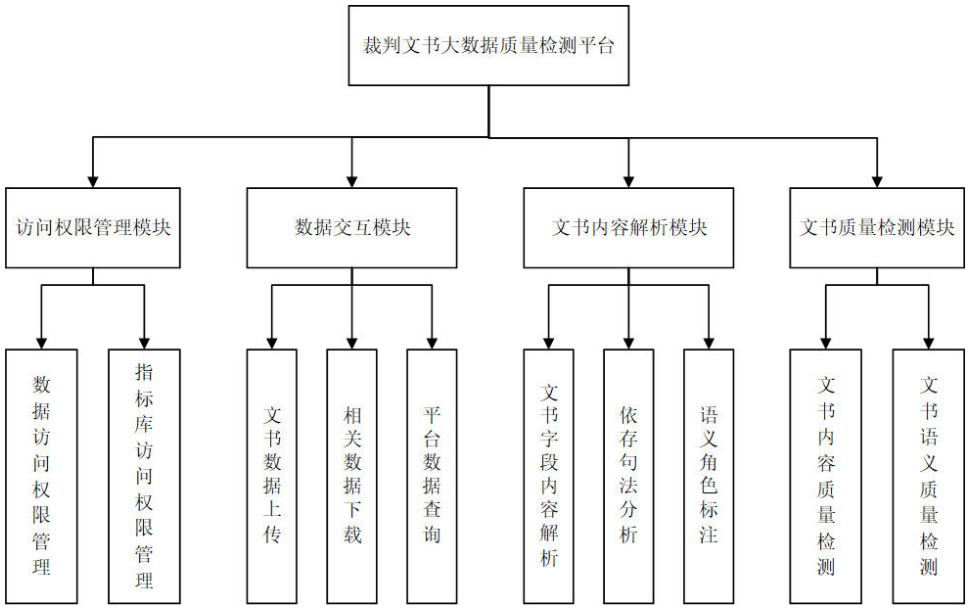

图 3.11: 平台功能模块图

3.6　系统详细设计

本节针对平台的访问权限管理模块、数据交互模块、文书内容解析模块和文书质量检测模块进行详细设计。

3.6.1　权限管理模块设计

访问权限管理模块负责用户可访问平台内容的范围。可划分为两个子模块：数据访问权限控制模块和指标设计认证模块。这两个子模块功能如下：

1. 数据访问权限控制模块对平台的HDFS，YARN，Hive，HBase进行细粒度的权限管理，使用Apache Ranger集中式安全管理框架控制开发者的访问权限，并审计其访问行为。

2. 指标设计认证模块管理质量指标的设计权限，为了使文书数据质量指标体系与时俱进，为开发者赋予了质量指标动态增删权限。指标名记录在Hive中，只有存储在Hive中的指标才能被平台调用。开发者只有获取指标设计权限，才能将设计的指标入库。这样的方式保证了指标有效性，不会因为开发者个人喜好而增添过多无意义指标。所有新增指标均需通过人为确认后，方可添加。

RangerPlugin是Apache Ranger的重要组件，在安装后以插件的形式与HDFS、Hive、HBase绑定，Ranger Admin通过此插件控制用户对数据的访问权限。因此，实现并调用RangerPlugin类是Hadoop组件权限控制的关键。

基于Apache Ranger的数据访问权限控制模块类图如图 3.12所示。

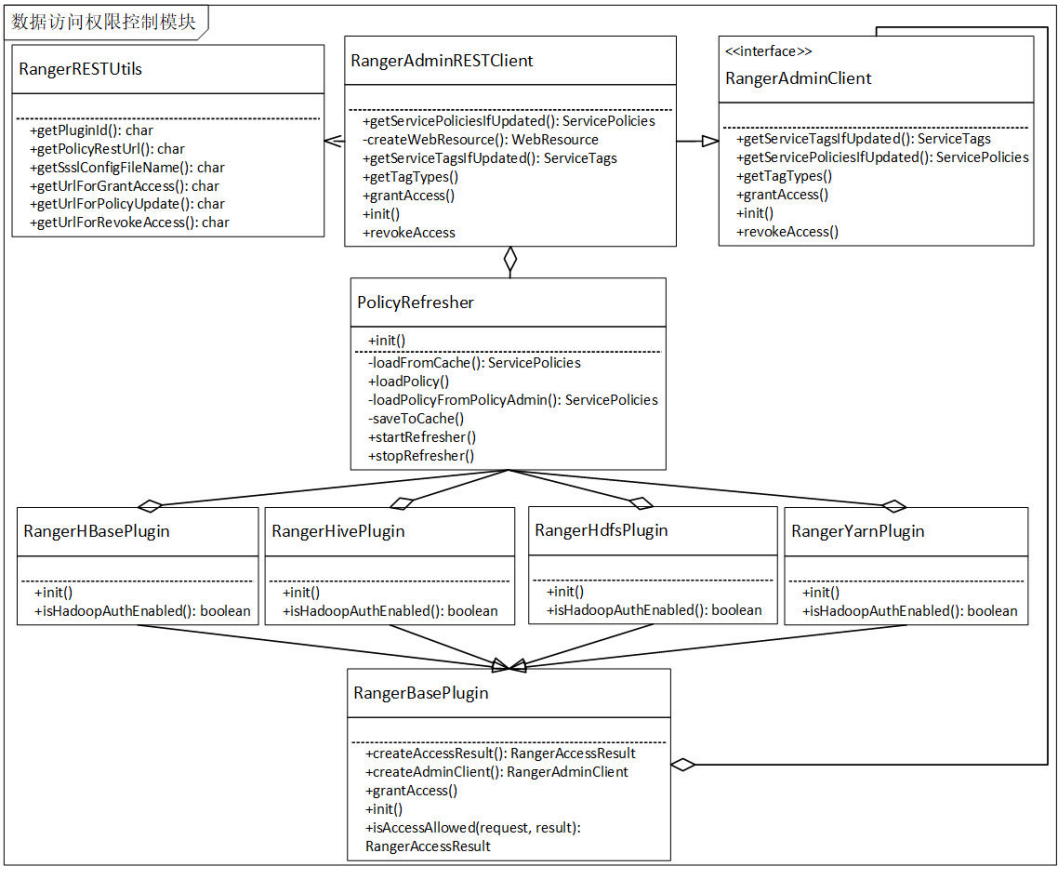

图 3.12: 数据访问权限控制模块类图

Ranger控制HDFS、HBase、Hive、YARN的插件类分别是RangerHdfsPlugin、RangerHBasePlugin、RangerHivePlugin、RangerYarnPlugin。Ranger Plugin定期轮询Admin获取权限控制策略。

指标设计认证模块类图如图 3.13所示。用户接入时发送认证请求，请求首先经过AuthController，将认证信息交由对应的服务处理。个人身份认证和密钥认证接口对应VerifyIdService和VerifyKeyService类。

AuthenticationService负责管理认证信息。管理员具有最高优先级的认证信息操作权限，可以通过AuthenticationManager进行用户权限管理。普通用户在权限认证通过后，可以使用DataDimensionDao类中提供的方法，在数据库中管理指标名。

在对文书进行质量检测时，需要从命令中解析指标名，只有库中存在的指标名，才能被平台调用。这样做保证了平台指标的统一性，提高了可维护性。

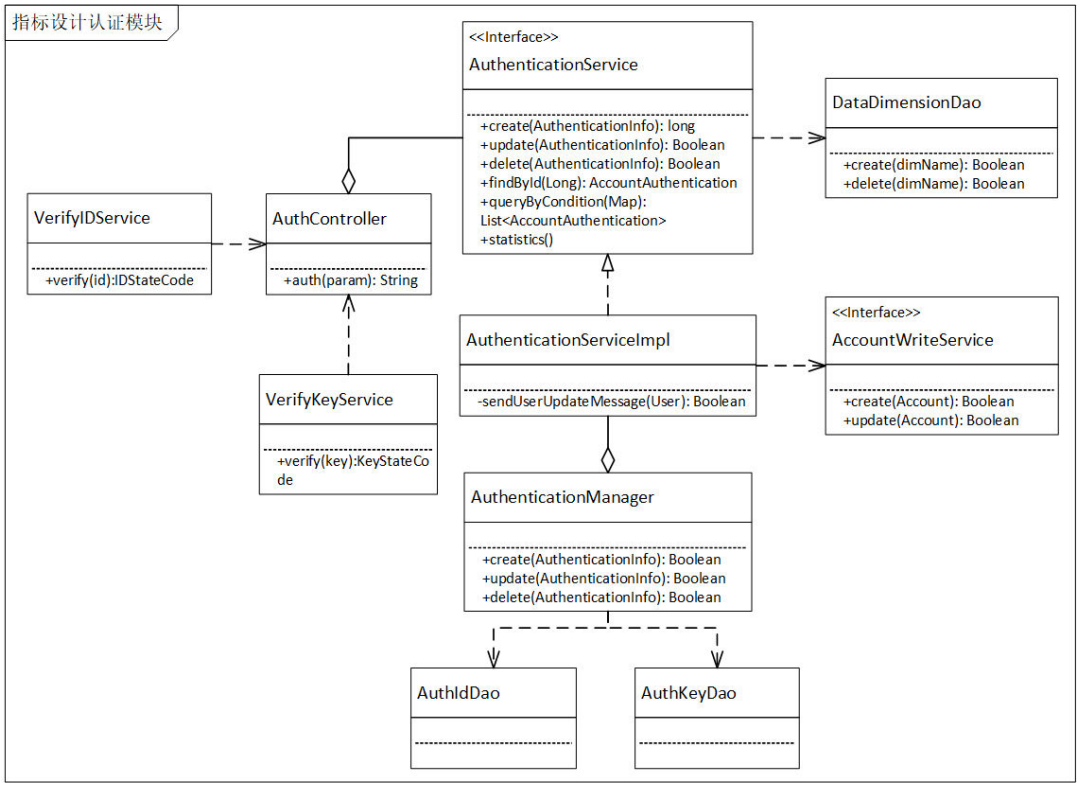

图 3.13: 指标设计认证模块类图

3.6.2　数据交互模块设计

数据交互模块负责用户与平台间裁判文书的数据交互。用户以30篇裁判文书为一个单位上传至平台，打包为zip格式。平台以Json文件格式为用户返回裁判文书的质量检测结果，内容中包含裁判文书原文、文书解析结果和文书质量检测结果。

如图 3.14所示，保证用户具有数据访问权限，用户数据访问请求首先要通过AccessFilter验证，AccessFilter实现了AccessRequestService类。AccessFilter通过DataInteractionService接口实例为用户提供数据交互功能。数据解压功能由DataStorageServiceImpl实现。UnstructDataDao和StructDataDao提供了结构化和非结构化数据的数据库操作，非结构化数据使用HBase的存储方式，结构化数据使用Hive的存储方式，二者均以HDFS作为底层架构，每一篇裁判文书以案号caseNum变量作为唯一标识。

NodeInfoService类为数据操作类的实例提供集群信息，如节点存储空间、HDFS文件存储位置等信息。

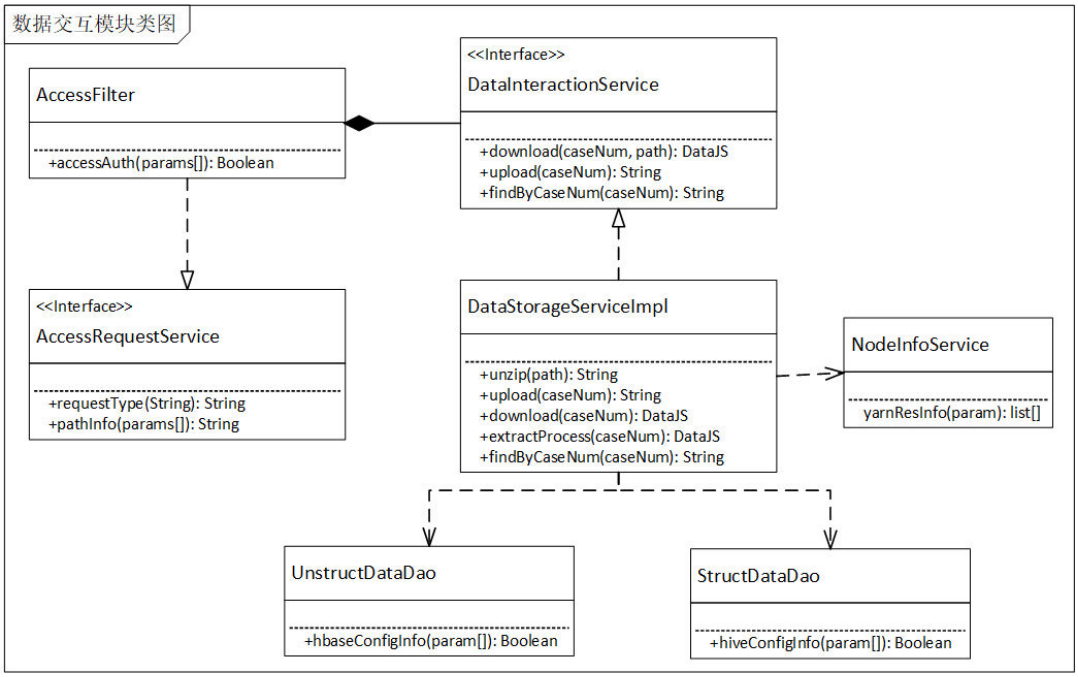

图 3.14: 数据交互模块类图

3.6.3　文书解析模块设计

文书解析模块负责对用户上传的文书内容进行解析，包括文本解析、字段抽取和句法及语义分析。

如图 3.15所示，文本解析类TextContentParse递归获取xml文书内容和格式特征，抽取案件详情后，经ContentStructurization标准化后存储至HBase和Hive，同时将xml原文也存储至集群中。TextFieldParse抽取关键字段，该模块获取案件关键信息，抽取案号、案由、罚金、引用法条等关键信息。该方法为文书内容质量检测提供数据基础。

DependencyParsing和SemanticRoleLabeling使用文书的“全文”字段进行依存句法分析和语义角色标注。“全文”字段包括案件的主体信息，是裁判文书的原文，使用自然语言描述。标注结果经格式标准化后存储至数据库，为语义质量检测提供数据基础。

句法及语义分析模块使用LTP工具首先对非结构化的案件详情文本进行分词和词性标注，然后进行依存句法分析和语义角色标注，最后将标注结果存储至HBase，为语义质量检测提供数据基础。语义角色标注依赖分词、词性标注和依存句法分析的结果。中文分词和词性标注都在DependencyParsing中完成，分词和标注结果作为重要的中间产物，也要存储至HBase。

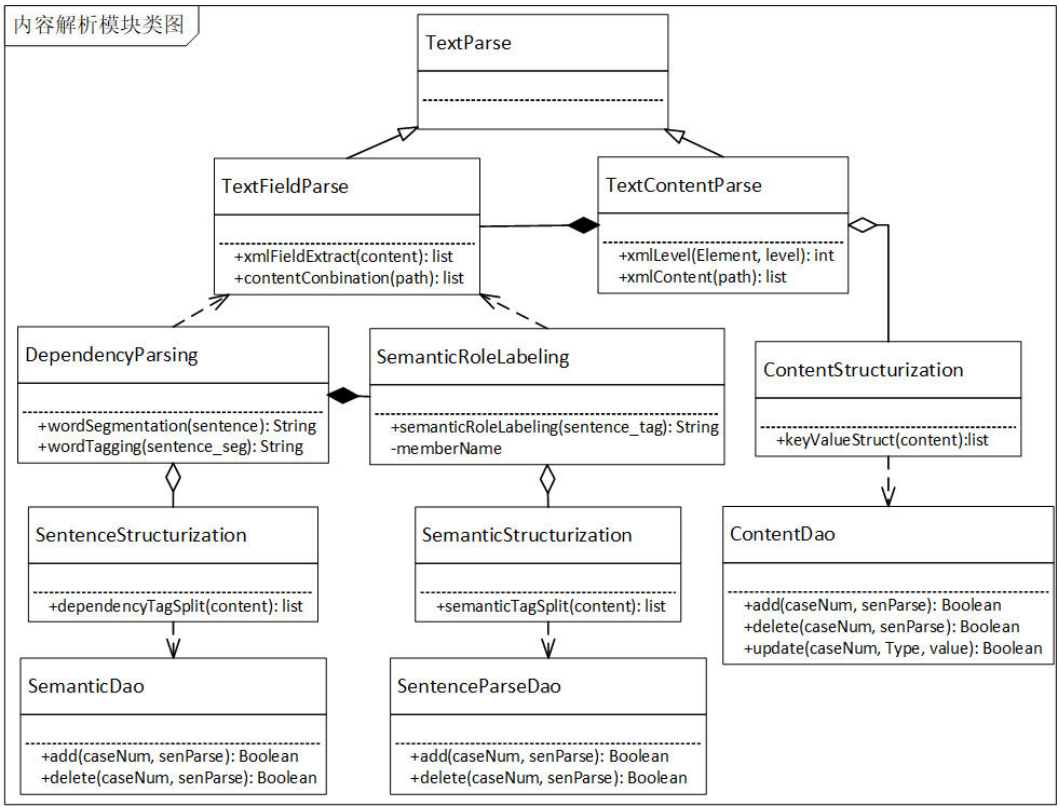

图 3.15: 文书解析模块类图

3.6.4　质量检测模块设计

质量检测模块完成裁判文书质检指标计算任务。在3.7.1中提到，可调用的数据质量指标保存在数据库中，权限管理模块保证了指标库的合理性。用户请求调用质量计算接口，需要通过指标验证，如图 3.16所示，DimisionVerifyImpl实现了DimisionVerifyService接口，在库中查询是否存在该指标。本模块使用抽象工厂模式提供可扩展的指标计算服务。DataQuality为质量指标接口，所有指标均需实现该接口，此部分通过依赖倒置原则解耦了指标的定义与实现，可以动态扩展质量指标。DQStore定义了指标设计原则，地方法院获取到指标设计权限后，可在CourtDQ中设计满足本地场景的质量指标。

质量计算由CalculationDim完成，最后保存在Hive数据库中。虽然本文提出的文书质量指标分为内容质量指标和语义质量指标，但它们的不同主要体现在理论定义上。其计算过程中使用到的数据均为非结构化文本类数据，具有一定相似性，因此将其相似部分抽象出来，由DataQuality接口定义其基本结构。

语义质量指标的计算需要使用3.7.3中语义标注的结果，内容质量指标的计算要使用依存句法分析和文本解析的结果。具体使用过程由指标类定义。

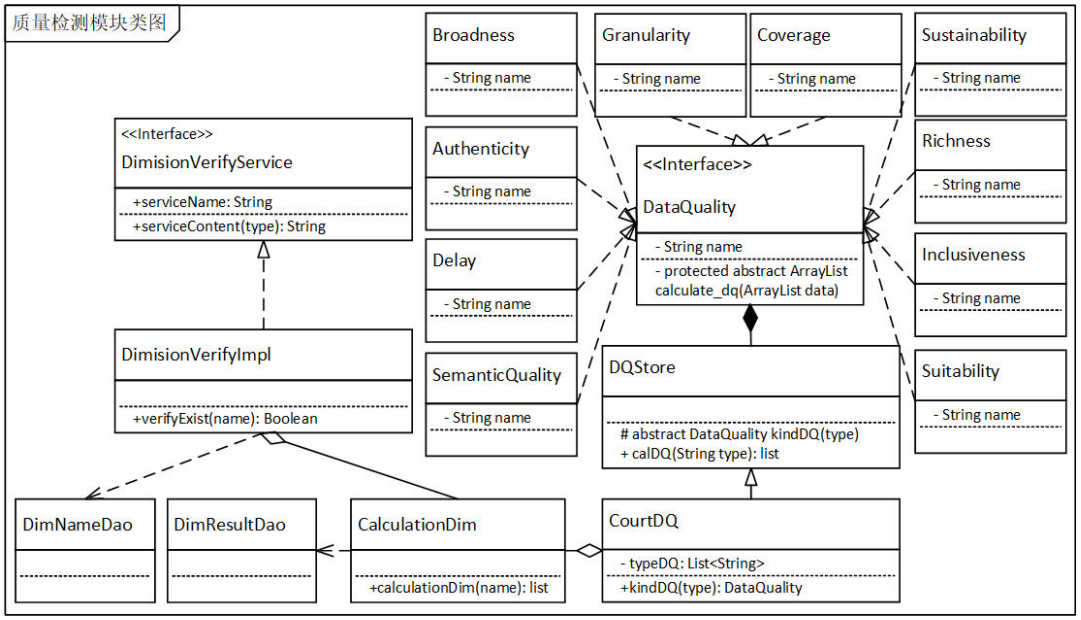

图 3.16: 质量检测模块类图

3.7　数据库设计

本节介绍平台核心数据库结构。文书质量检测平台需要存储文书原文本、文书解析内容和质量检测结果等信息，涉及到非结构化、半结构化和结构化数据。针对非结构化和半结构化数据，采用HBase分布式数据库，针对结构化数据，采用Hive分布式数据库。

3.7.1　关系型数据库设计

图 3.17和图 3.18是平台主要关系型数据库ER图。

图 3.17表达了文书数据存储库的存储关系。平台的数据库设计围绕案件表case展开。案件文书信息可分为内容信息和结构信息，case表存储文书重要字段的文本信息，如案号、案由等，此表用于提供案件关键字段查询服务。doc_struct 表存储文书结构信息，与case表为一对一关联关系。doc_struct表中的结构信息主要用于文书内容质量度量，涉及xml元素的“广度”和“深度”。文书元素广度是被度量元素内部包含的二级元素个数，元素深度是被度量元素可展开的最大深度。

除了要记录文书内容之外，case_info表记录了文书平台信息，包括文书上传时间、上传用户id、文书数据所在节点id等信息。文书质量需要责任到人，为了追踪文书来源，限定司法人员只能访问本人上传的数据，需要记录每篇裁判文书的上传者。

同时，id_server记录了文书在集群中的存储节点，与server_info表相关联，帮助获取集群节点的状态信息。而case_quality表记录了裁判文书数据质量的检测值。

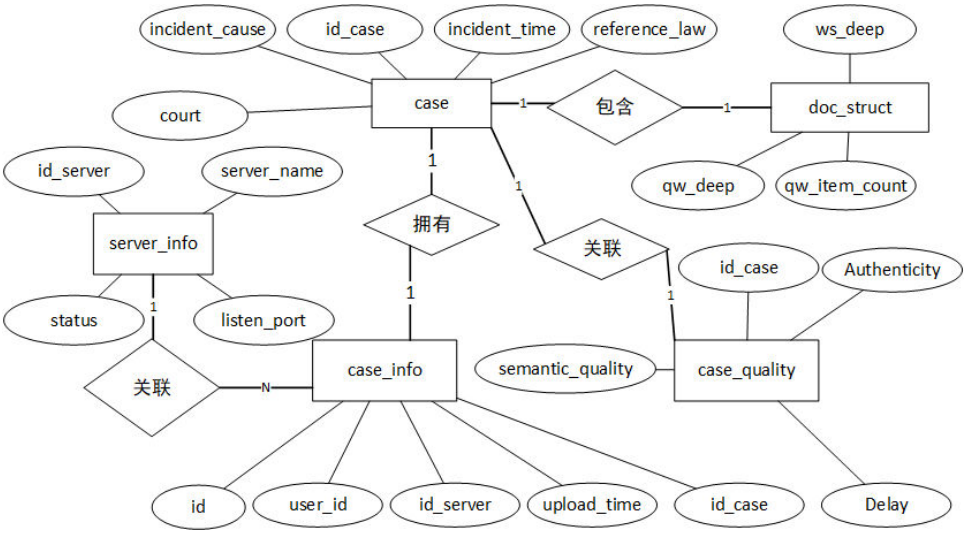

图 3.17: 关系型文书数据库实体关系图

图3.18是用户指标设计权限管理的ER图。用户账号存储在账号表account中，user_id是用户身份唯一标识，用户开发机ip作为密钥key。平台接到用户指标管理请求时，先验证用户权限permission，只有permission值为1时，才能向指标表dimension添加质量检测指标。指标表除了记录指标名，还记录了设计它的user_id。

数据访问权限由Ranger插件管理。

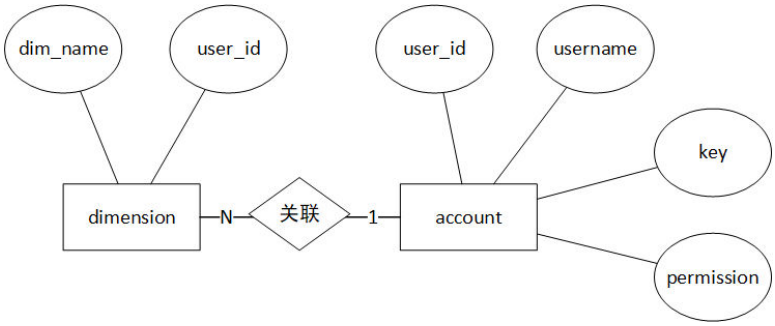

图 3.18: 用户指标设计权限实体关系图

数据库表结构设计如下所述：

(1) 案件表case：case表中存储了易于结构化的案件关键信息，包括案

号id_case，案由incident_cause，案发时间incident_time，引用法条reference_law，起诉人plaintiff，经办法院court。

表 3.9: 案件表case表结构

字段名	类型	允许为空	说明
id_case	bigint(20)	否	案号(文书唯一标识)
court	varchar(64)	否	经办法院
incident_cause	varchar(60)	否	案由
incident_time	datetime	否	案发时间
reference_law	varchar(64)	否	引用的法律条文
plantiff	varchar(64)	否	主要涉案人

(2) 案件结构表doc_struct：在3.6.3节描述的文书解析方法，可以解析出文书的结构化信息，该结构化数据存储在doc_struct表中。表中存储了文书重点度量元素的结构信息，每个元素都具有“深度”和“广度”两个值。

表 3.10: 案件结构表doc_struct表结构

字段名	类型	允许为空	说明
id_case	bigint(20)	否	案号
qw_deep	bigint(10)	否	“全文”元素最大深度
qw_item_count	bigint(10)	否	“全文”内元素总数
ws_deep	bigint(10)	是	“文首”元素最大深度
ws_item_count	bigint(10)	是	“文首”内元素总数
dsr_deep	bigint(10)	是	“当事人”元素最大深度
dsr_item_count	bigint(10)	是	“当事人”内元素总数
ssjl_deep	bigint(10)	是	“诉讼记录”元素最大深度
ssjl_item_count	bigint(10)	是	“诉讼记录”内元素总数
ajjbqk_deep	bigint(10)	是	“案件基本情况”元素最大深度
ajjbqk_item_count	bigint(10)	是	“案件基本情况”内元素总数
cpfxgc_deep	bigint(10)	是	“裁判分析过程”元素最大深度
cpfxgc_item_count	bigint(10)	是	“裁判分析过程”内元素总数
pjjg_deep	bigint(10)	是	“判决结果”元素最大深度
pjjg_item_count	bigint(10)	是	“判决结果”内元素总数
ww_deep	bigint(10)	是	“文尾”元素最大深度
ww_item_count	bigint(10)	是	“文尾”内元素总数

(3) 案件平台信息表case_info：case_info存储平台相关的文书信息，包括文书上传者user_id，文书所在服务器名server_name 和上传时间upload_time。

表 3.11: 案件系统信息表case_info表结构

字段名	类型	允许为空	说明
id	bigint(20)	否	文书存储信息编号
id_case	bigint(20)	否	案号
user_id	int(10)	否	上传此文书的用户id
server_name	varchar(64)	否	所在节点名
upload_time	datetime(8)	否	上传时间

(4) 集群节点信息表server_info：case_info表和server_info表是关联关系，server_info表中保存了文书存储的节点名server_name，节点状态status，消息监听端口号listen_port。

表 3.12: 集群节点信息表server_info表结构

字段名	类型	允许为空	说明
id_server	bigint(20)	否	集群节点编号
server_name	varchar(64)	否	服务器节点名
status	tinyint(1)	否	节点状态（1为启用、0为故障、-1为锁定、-2为删除）
listen_port	bigint(10)	否	该节点消息监听端口号

(5) 质量结果表case_quality：存储文书各指标计算结果。文书以案号为唯一标识，仅通过案号与其他表产生关联。新指标入库时，此表也要同步更新。

表 3.13: 文书质量度量结果表case_quality表结构

字段名	类型	允许为空	说明
id_case	bigint(20)	否	案号
semantic_quality	float(10, 5)	否	语义质量
delay	float(10, 5)	否	延迟性
authenticity	float(10, 5)	否	真实性
broadness	float(10, 5)	否	广阔性
granularity	float(10, 5)	否	细致性
coverage	float(10, 5)	否	遍及性
sustainability	float(10, 5)	否	持续性
richness	float(10, 5)	否	丰富性
indusiveness	float(10, 5)	否	包容性
suitability	float(10, 5)	否	适配性

(6) 账号表account：表中存储了用户权限信息，user_id是用户身份唯一标识，用户开发ip作为密钥key，permission是其权限标识，permission只有为1时，用户才具有指标设计权限。

表 3.14: 账号表account表结构

字段名	类型	允许为空	说明
user_id	bigint(10)	否	用户身份标识
username	varchar(20)	是	用户名
key	varchar(20)	是	密钥
permission	tinyint(10)	否	权限（为1代表具有管理指标权限，为0无权限）

(7) 指标表dimension：存储可被调用的指标名。计算数据质量前，要在该表中查看是否存在指标。并且该表中保存了指标的设计者user_id。

表 3.15: 质量指标表dimension表结构

字段名	类型	允许为空	说明
dim_id	bigint(10)	否	指标id
dim_name	varchar(20)	否	指标名
user_id	bigint(10)	否	设计指标的用户名

3.7.2　非关系型数据库设计

裁判文书用自然语言进行描述，中文是典型的非结构化数据，xml格式的文书是<元素，元素值>格式，每一个“元素”都代表文书的一个字段，“元素值”代表文书在此字段下的具体内容，例如<案由，盗窃罪>。不同裁判文书的字段类型不同，因此不适合使用关系型数据库进行存储。同时，用户与平台交互、数据质量计算等平台功能，均需保存和读取xml格式的文书数据。HBase是面向Hadoop的数据库，它利用Hadoop的HDFS作为其文件存储系统，适合存储非结构化数据。

HBase表是多维稀疏表，表中的数据是字符串，没有数据类型。表中每一行都有一个行键，被分为许多列族的集合，支持动态扩展，无须事先定义便可添加一个列族或者列，所有列以字符串形式存储。HBase中涉及到三个重要的概念，行键(Row Key)、列族(Column Family)和列(Column)，以及时间戳(Time Stamp)。

行键：HBase一张表中可以存储上亿条记录，每一行都由一个关键字row key来唯一标识，并对其按照字典序排序存储。

列族和列：HBase中的每一列都属于某个列族。列族需要在使用表之前定义，而列不需要事先定义。列族数量不能太多，列名以列族作为前缀。例如：（判决结果：是否退还案件诉讼费），（判决结果：再审复查结案方式）都属于“判决结果”这个列族。每个列族内可以存放许多列，并且各列族内列的数量可以不同。

时间戳：HBase通过关键字、列和时间戳的三元组来唯一确定一个存储单元。每个存储单元都保存着同一份数据的多个版本，不同版本间的数据按照时间倒序排序。版本通过时间戳来索引。时间戳可以由HBase在写入时赋值。

基于以上讨论，结合平台需求和裁判文书结构，设计了以下HBase表。

(1) CaseDoc表按照字段存储文书文本内容，如表 3.16所示。文书内容分为正文和附加段两部分，在表中构成了text_info和additional_segment两个列族。列族根据文书内容特征，划分为具体的属性列。

正文text_info列族包含了以下属性列：文书全文full_text、文首head、当事人litigant、诉讼记录litigation_record、案件基本情况basic_situation、裁判分析过程referee_process、判决结果judgment和文尾tail。附加段additional_segment列族包含了审判类别case_category 和审判程序trail_procedure_category

表 3.16: 文书内容信息表CaseDoc表结构

表名	行键	列族	列
CaseDoc	CaseNum	text_info:	full_text head litigant litigation_record basic_situation referee_process judgment tail
		additional_segment:	case_category trial_procedure_category

(2) 除了记录文本原始内容，平台也对依存句法分析及语义角色标注结果、句子贡献度进行了记录，如表 3.17所示。文本分析方法面向全文句子，sentence_dependency_parse 和sentence_semantic_label是文书全文段句子的句法及

语义分析标注结果，sentence_contribute中记录每句话的贡献度，以空格作为分隔符。

表 3.17: 句法及语义解析结果表Analysis表结构

表名	行键	列族	列
Analysis	CaseNum	sentence_contribution:	sentence_dependency_parse sentence_semantic_label sentence_contribute

3.8　本章小结

本章首先分析了裁判文书数据质量指标体系的构建方法，设计了各指标的理论模型，然后界定了文书数据质量检测平台的系统边界，并从数据交互、文书内容解析、质量检测和平台权限设计的角度进行需求分析，描述了平台基本功能。并且详细设计了平台各功能模块的类图结构和数据库存储方法，为平台实现打下坚实的理论基础。

第四章　系统实现

本章介绍了裁判文书大数据质量检测平台中文书解析、质量检测、数据交互和权限管理模块的核心实现，以及Hadoop的高可用分布式配置；根据各模块特点给出实现环节的流程图和组件图，分析并展示了文书质量检测结果。最后对平台核心功能进行了测试。

4.1　文书解析

本节提取了裁判文书的关键信息，为文书质检做铺垫。文书解析建立在xml格式的裁判文书基础上，本节首先分析文书结构，确定重要信息段，然后对文书字段进行解析，最后针对解析出的文书原文进行自然语言解析。

4.1.1　文书结构分析

裁判文书记录了案件审判执行的详细信息，是司法公正的载体，是法官向当事人提交的“答卷”。为此，裁判文书中记录了罪名、案由、当事人、经办法院、判决结果、引用法条等信息字段。不同案由的裁判文书之间信息字段略有不同。

裁判文书按照案件裁判方式可分为5种，包括判决书、裁定书、调解书、决定书和通知书；按照案件类别可分为民事、刑事、行政、国家赔偿与司法救助等10类，其中民事、刑事和行政三类案件较多。

本文分析了刑事、民事、行政三大类案由的文书特点，以刑事类二审裁判文书为例，表 4.1列举了部分可从文书中提取的重要字段。

表 4.1展示了文书中的可提取元素，本文将其组织成三层。下层是对上层元素的细粒度分解。首层“全文”以非结构化自然语言的形式进行描述，用于文书语义解析；中层元素包含文首、当事人、诉讼记录、案件基本情况、裁判分析过程、判决结果、文尾和时间，将“全文”拆分为8项，每一项记录其层数；下层元素是对中层元素具体信息的描述。

4.1.2　文书字段解析实现

平台使用dom4j工具解析xml，除了获取xml的元素信息之外，还需要获取xml的结构信息。解析结果为文书自然语言处理和内容质量指标的计算提供数据支撑。具体实现如图 4.1 所示。

表 4.1: 文书可提取字段信息表

首层元素	中层元素	下层元素	中层元素	下层元素
全文	文首	文书制作单位 法院文书种类 经办法院 案号 案件类别 文书种类	当事人	公诉方 起诉方 诉讼参与人 国籍 强制措施 自然人身份
	诉讼记录	审判组织 起诉主案由 罪名代码 案件由来与审理经过	案件基本情况	质证情况 同案犯们 被害人 本审审理段
	裁判分析过程	量刑情节 法律法条引用	判决结果	二审结案方式 刑事判决结果分组
	文尾	裁判时间 审判组织成员	...	... ...

文书内容解析TextContentParse：包括文书结构信息解析和文书字段信息解析。为了获取文书结构信息，利用xml树状结构，采用递归算法，用参数level获取当前层数，计算文书中层元素和下层元素的节点个数及深度，返回计算结果；平台关注文书下层元素的具体内容，其代表文书的细节信息。

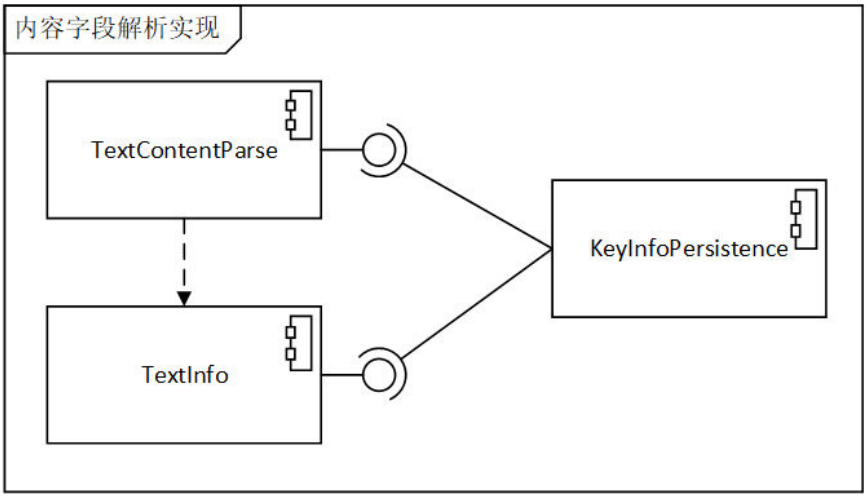

图 4.1: 文书字段解析实现

文书基本信息TextInfo：保存了文书原始数据和平台信息，是文书处理流程中的信息集合。

文书信息持久化KeyInfoPersistence：将文书基本信息、文书结构信息和文书字段信息使用Redis持久化到集群节点node的内存中，为质量检测和自然语言处理做铺垫，减少数据存取时间。

图 4.2是本模块中重要的文书递归解析核心代码。

```
public class TextContentParse{
    public static int parserNode(Element ele,int level){
    …
        if(level==3){
            thirdnodes++;
            List<Element> eleList = ele.elements();
            //递归遍历父节点下的所有子节点
            for(Element e : eleList){
               if((temp=parserNode(e,level+1))>max){
                    max=temp;
                }
            }
            thirdnodeslevels+=max+1;
            return max + 1;
        }
        if(ele==null) return 0;
        List<Attribute> attrList = ele.attributes();
        List<Element> eleList = ele.elements();
        for(Element e : eleList){
            if((temp=parserNode(e,level+1))>max)
                max=temp;
        }
        return 1+max;
    }
    …
}
```

图 4.2: 文书递归解析核心代码

4.1.3　文书语义解析实现

文书语义解析模块负责平台对文书的自然语言处理部分，为语义质量检测提供数据支撑。本模块对文书字段解析结果中的“全文”字段进行语义分析。全文字段是裁判文书原文，由法官按照规定格式书写，符合中文语言习惯，并且词语具有上下文关系。本模块分析句子之间的语义关系，在词语级别进行语义标注，为文书的每一个词赋予其句法和语义标签。

本模块语义解析流程如图 4.3所示。

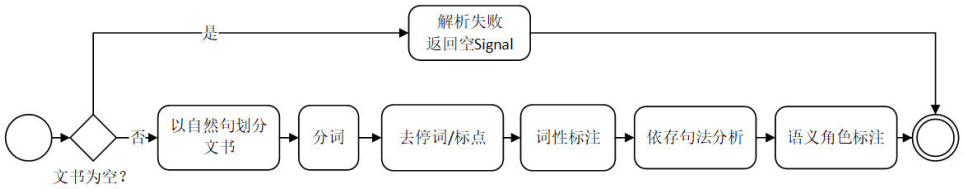

图 4.3: 文书语义解析实现

本流程中，后一阶段的计算依赖于前一阶段的正确执行。依存句法分析和语义角色标注是流程核心，平台返回的标注结果利用LTP语言云平台可视化后如图 4.4 所示。文书语义特征主要根据此阶段的标注结果进行计算。

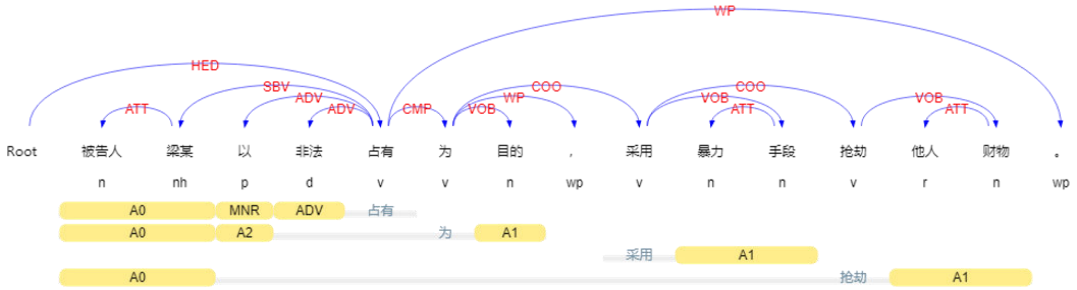

图 4.4: 文书语义标注结果示例

依存句法分析和语义角色标注使用哈工大语言技术平台的python-LTP工具实现。依存句法分析面向经过分词和词性标注的文书文本，图 4.5是核心代码，首先加载LTP模型，然后将分词和词性标注的结果作为输入，最后输出句法分析的标注结果。

```
def dependency_parser(self, list_word_segmentation, list_postags):
    par_model_path = os.path.join(self.LTP_DATA_DIR, 'parser.model')
    parser = Parser() #  初始化实例
    parser.load(par_model_path) #  加载模型
    for i in range(len(list_word_segmentation)):
        arcs = parser.parse(list_word_segmentation[i], list_postags[i])
        list_arc_temp = []
        for arc in arcs:
            list_arc_temp.append(arc.relation)
        list_arcs_content.append(list_arc_temp)
        list_arcs.append(arcs)
    parser.release()
    return list_arcs, list_arcs_content
```

图 4.5: 依存句法分析核心代码

语义角色标注与依存句法分析类似，其输入是文书文本的分词、词性标注和句法分析结果，最终返回语义角色标注结果，核心代码如图 4.6所示。

```
def sementic_role_label(self, list_word_segmentation, list_postags, list_arcs):
    list_roles = []
    srl_model_path = os.path.join(self.LTP_DATA_DIR, 'pisrl_win.model')
    labeller = SementicRoleLabeller()
    labeller.load(srl_model_path)
    for i in range(len(list_word_segmentation)):
        roles = labeller.label(list_word_segmentation[i], list_postags[i], list_arcs[i])
        list_roles.append(roles)
    return list_roles
```

图 4.6: 语义角色标注核心代码

4.2　文书质量检测

本节介绍了文书质量检测的具体实现，包括文书内容质量检测和文书语义质量检测。平台以接口方式为开发者提供文书质检服务，本节随机抽取2010年至2015年天津地区的刑事案件裁判文书共计8541 篇，经平台质量检测后，对其返回数据进行可视化，展示数据质量的检测结果，并根据统计结果确定了各指标阈值。

4.2.1　内容质量检测实现

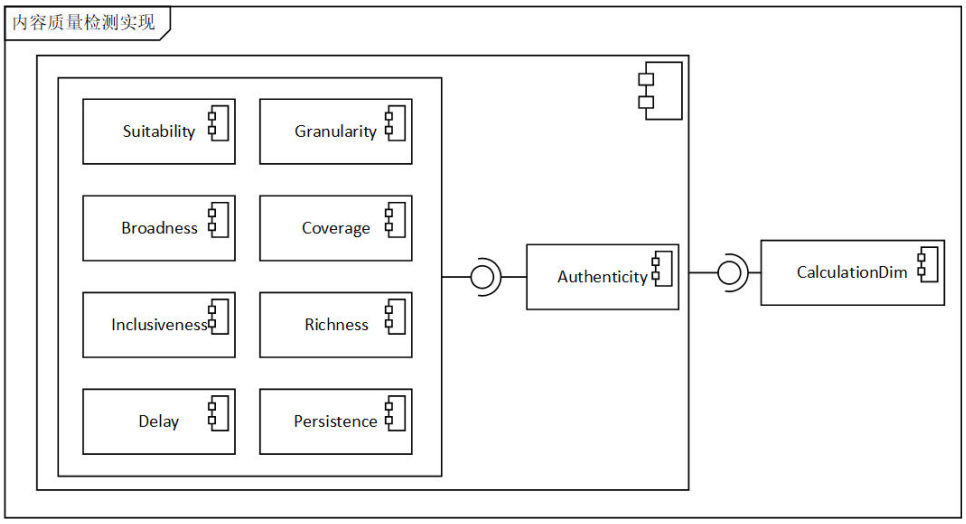

图 4.7: 文书内容质量检测实现

本模块检测文书在适配性、广阔性、细致性、遍及性、延迟性、持续性、包容性、丰富性和真实性九个维度的表现。

文书内容质量检测依赖于文书字段解析结果，关注文书下层元素的具体内容。各检测指标以组件形式集成，可灵活增添。各指标检测结果以接口方式向DataQuality集成，如图 4.7所示。

基本监测指标有8个，Authenticity是基本监测指标的加权和。这些指标的实现继承自DataQuality，向CalculationDim模块提供监测接口。接下来，本文随机选取2010年至2015 年天津地区刑事案件裁判文书共计8541篇，构成待检测数据集，在平台上模拟监测文书数据质量。数据集信息如表 4.2所示。

表 4.2: 待检数据详情表

项目	详情
文书总数	8541
裁判文书类型	判决书，裁定书
罪名	盗窃罪，抢劫罪，交通肇事罪，破坏交通设施罪，违法发放贷款罪
涉及法律条文总数	247
审判环节	刑事二审

首先将裁判文书以30篇为一组打包，然后调用平台upload接口，将数据上传至平台。平台自动将数据分配至集群节点，在服务器上解析数据包，检测文书质量并以Json格式返回度量结果。平台检测接口交互如图 4.8所示：

```
~ node01$ upload cases.zip
Uploading...
| ######################################################################## | 100%
Allocating servernode...
Success
Natural language processing...
Success
Data quality inspecting...
Success
Generating quality test result file...
Success
```

图 4.8: 文书质量检测运行图

表 4.3展示了检测结果，并使用Kolmogorov-Smirnov方法检验数据是否符合正态分布。当Kolmogorov-Smirnov正态性检测值大于0.05时，认为向量服从正态分布。由表可知，两个时间相关的检测指标，延迟性和持续性不服从正态分布，其余指标均较为符合正态分布规律。针对符合正态分布的前6个指标，采用

直方图对其频率进行可视化，如图 4.9所示。X轴为指标值，范围为[0, 1]，步长为0.00001，Y轴是该指标值范围内的文书数量。

表 4.3: 文书内容质量检测结果表

	适配性	广阔性	细致性	遍及性	包容性	丰富性	延迟性	持续性	真实性
$case_{01}$	0.7153	0.5783	0.6977	0.2444	0.4712	0.6916	0.0034	0.0004	0.5629
$case_{02}$	0.7372	0.6189	0.7458	0.1778	0.5281	0.7277	0.0025	0.0004	0.5798
$case_{03}$	0.7591	0.6498	0.7119	0.1662	0.5656	0.7229	0.0046	0.0002	0.5874
$case_{04}$	0.6788	0.7406	0.6286	0.114	0.8361	0.6369	0.001	0.0006	0.5806
$case_{05}$	0.8029	0.5977	0.7213	0.2966	0.4132	0.7518	0.0008	0.002	0.6037
$case_{06}$	0.635	0.6653	0.6667	0.144	0.7338	0.6317	0.0	0.1813	0.5915
$case_{07}$	0.8832	0.5682	0.86	0.4406	0.2655	0.8655	0.0088	0.0003	0.6641
$case_{08}$	0.6569	0.5595	0.8511	0.1971	0.498	0.736	0.0018	0.0006	0.5649
$case_{09}$	0.8978	0.7558	0.6087	0.2097	0.6245	0.7479	0.01	0.0001	0.6456
$case_{10}$	0.8321	0.6299	0.7368	0.2483	0.4452	0.7757	0.0098	0.0001	0.6154
$case_n$	...	...	...	...	...	...	...	...	...
mean	0.7690	0.6472	0.7264	0.2158	0.5495	0.7356	0.0787	0.0056	0.6022
std	0.1147	0.0519	0.0749	0.0801	0.1549	0.0688	0.1892	0.0416	0.0353
kstest	0.6542	0.6701	0.3512	0.3341	0.4185	0.4677	0.0205	0.0094	0.7211

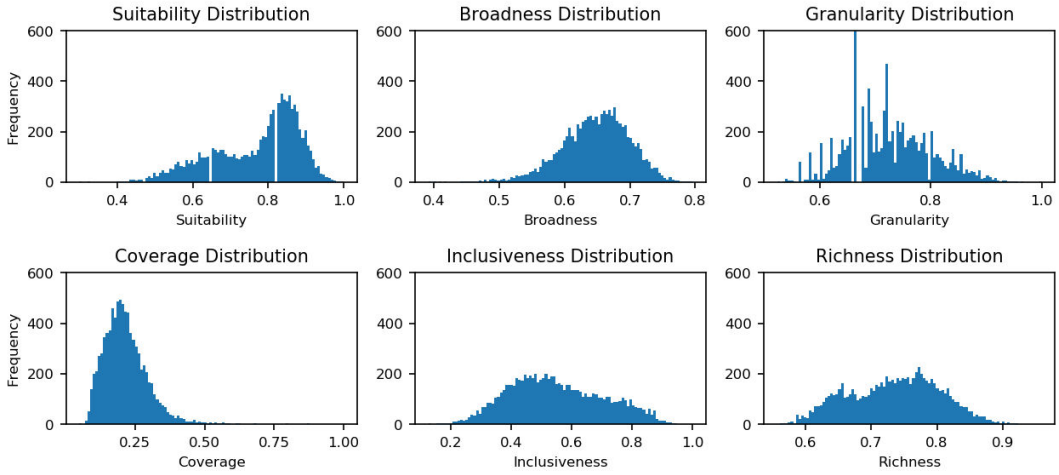

图 4.9: 指标频率统计直方图

对上述6个指标计算其正态分布的概率密度函数，概率密度如图 4.10所示。

Delay和Persistence不符合正态分布特性，本文使用散点图对其可视化，如图 4.11所示。对于延迟性，不同案件类型要求的判决时间不同，图中在一定程度上反映了案件的集中特征。对于持续性，法院针对大部分案件的判决在规定

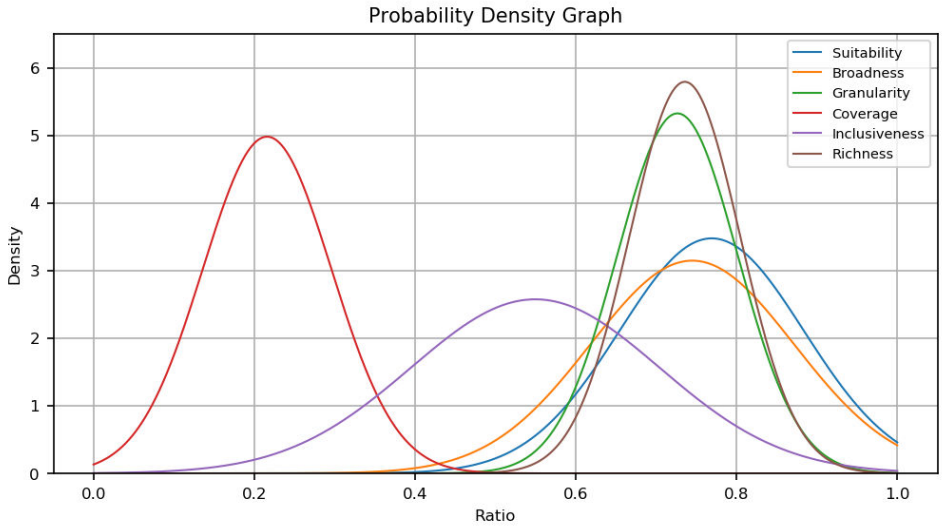

图 4.10: 指标概率密度图

时间范围内完成，导致数据分布在[0, 0.15]内呈递减下降的分布特征。在一定程度上表示法院希望尽量在较短时间内对案件作出判决。

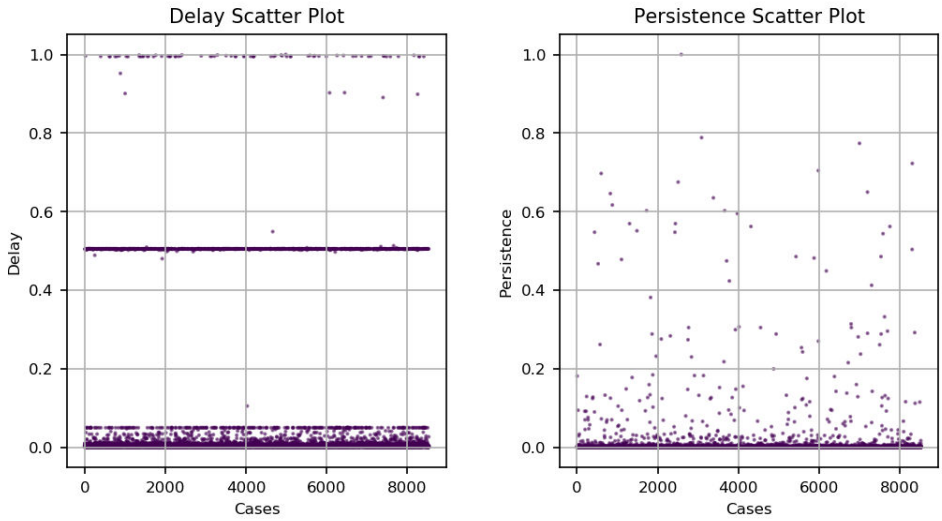

图 4.11: 延迟性和持续性散点图

裁判文书真实性是基础指标的加权平均值。其反映裁判文书忠于事实的程度，也反应了用户对各指标的重视程度，用来衡量数据集质量。本文设置所有指标权重为0.125，其符合正态分布规律。

图 4.12是真实性的频率分布图和散点图。

上述分析表明不同指标的分布不同，本文人工设定了质量阈值超参数。

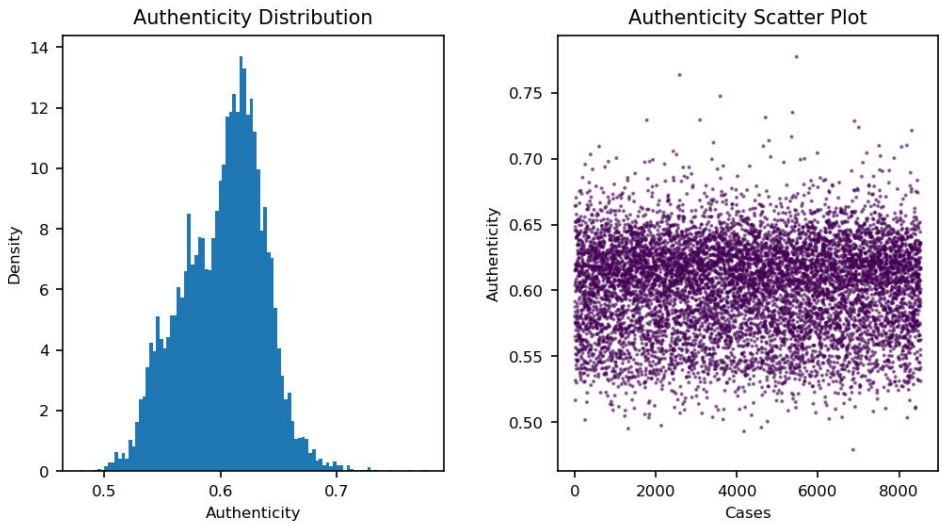

图 4.12: 真实性频率分布图和散点图

表 4.4 是各指标阈值和文书达标情况。

表 4.4: 文书质量达标率及阈值表

	适配性	广阔性	细致性	遍及性	包容性	丰富性	延迟性	持续性	真实性	语义质量
阈值	0.4732	0.5311	0.7264	0.6293	0.1201	0.2533	0.0008	0.0001	0.5501	0.4972
达标率	0.9066	0.8972	0.9285	0.8761	0.8959	0.8621	0.8837	0.9368	0.9241	0.9211

以细致性为例，图 4.13是细致性计算的核心代码。

```
public class Granularity() {
    public static Map<String, Double> Calculate(){
        Iterator iterFileNameThirdnodes = mapFileNameThirdnodes.entrySet().iterator();
        while(iterFileNameThirdnodes.hasNext()){
            Map.Entry entryFileNameThirdnodes = (Entry) iterFileNameThirdnodes.next();
            String fileName = (String) entryFileNameThirdnodes.getKey();
            // 获取文书结构信息
            int thirdnodes = Integer.parseInt((String) entryFileNameThirdnodes.getValue());
            int thirdnodeslevels = Integer.parseInt(mapFileNameThirdnodeslevels.get(fileName));
            // 执行计算
            double result = (double)thirdnodes/(double)thirdnodeslevels;
            mapGranularityResult.put(fileName, result);
        }
        return mapGranularityResult;
    }
}
```

图 4.13: 文书细致性度量核心代码

文书解析为xml文件后具有了树状结构，树深度在一定程度上反映文书对案件描述的细致程度。iterFileNameThirdnodes存储文书结构信息，依据细致性计算公式，返回计算结果mapGranularityResult<String, Double>，其中文书案号为String类型，文书细致性度量结果是Double类型。

4.2.2　语义质量检测实现

本模块检测文书语义质量，其质量检测方法依赖文书语义解析的结果。根据3.2.2小节所述，为了计算语义质量，首先要利用标注集计算文书的语义特征，然后计算句子的语义贡献度，最后根据公式导出文书语义质量检测结果。依据上述模块特征，图 4.14是语义质检的实现框架。

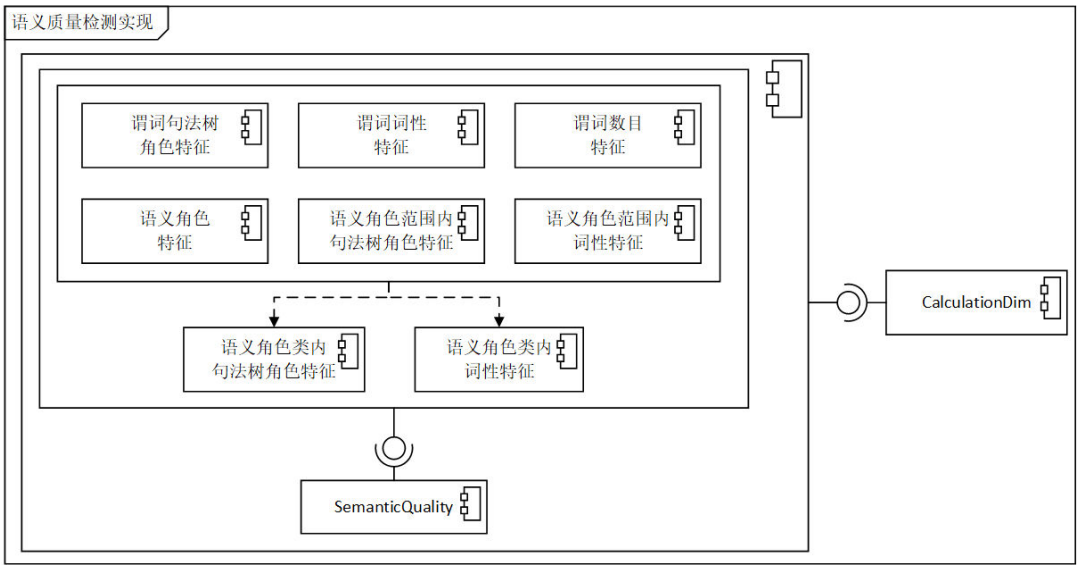

图 4.14: 文书语义质量检测实现

基础语义特征有6个，包括谓词句法树角色特征、谓词词性特征、谓词数目特征、语义角色特征、语义角色范围内句法树角色特征和语义角色范围内的词性特征，它们关注文本解析后语义标签的表层统计特征。

语义角色类句法树角色特征和语义角色类内词性特征使用语义角色标注结果，将句子按照短语标签分类，在每个簇中计算句法树特征和词性特征，其计算过程依赖基础语义特征。语义特征组件向语义质量指标SemanticQuality组件提供特征数据，最后以接口方式和质量指标计算组件CalculationDim集成。

上一小节返回的计算结果中包含了文书语义特征值和文书语义质量检测结果。表 4.5展示部分语义角色类句法树角色特征值。可以看到文本中的短语被按照语义角色标签分类，每一类中依据句法树解析结果给出了该类下的特征值。

表 4.5: 语义角色类句法树角色特征值表

语义类	句法标签	频率	特征值	语义类	句法标签	频率	特征值
施事	ATT	2303	0.3994	受事	VOB	2955	0.2118
	SBV	1975	0.3425		ADV	1316	0.0943
时间	ATT	656	0.3458	目的|原因	VOB	20	0.1123
	ADV	615	0.3241		WP	19	0.1067
动作开始	RAD	2	0.0400	动作结束	ATT	1	0.5000
	CMP	1	0.0200		VOB	1	0.5000

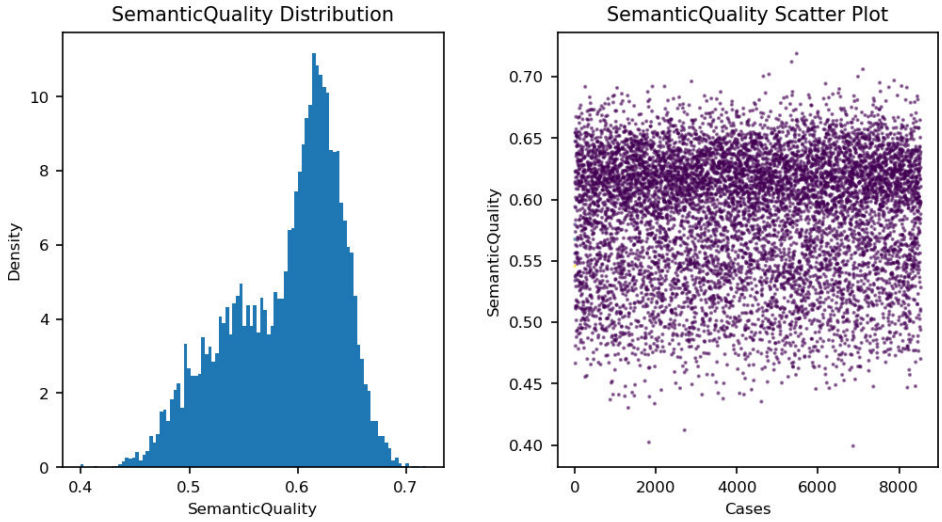

图 4.15: 文书语义质量分布

图 4.16以“语义角色类内句法树角色特征”为例，给出了核心代码。

```
dict_arg_ptgs_ratio = {}
for index_argptgs in range(len(datainfo.arg_ptgs_ratio)):
    arg_ptgs_ratio = datainfo.arg_argptgs_ratio[index_argptgs]
    arg = arg_ptgs_ratio[0]
    list_postags_ratio = arg_ptgs_ratio[1]
    dict_ptgs_ratio = {}
    for i in range(len(list_postags_ratio)):
        postag = list_postags_ratio[i][0]
        amount = list_postags_ratio[i][1][0]
        ratio = list_postags_ratio[i][1][1]
        dict_ptgs_ratio.update({postag: [amount, ratio]})
    dict_arg_ptgs_ratio.update({arg: dict_ptgs_ratio})
```

图 4.16: 语义角色类内句法树角色特征计算核心代码

返回的dict_arg_ptgs_ratio中保存了文书此类特征值。

```
def contribute_sen(self, dict_argarcsdict[seninx], dict_argptgsdict[seninx]):
list_sentence_result = []
for seninx, dict_roleinx_argarcsdict in dict_argarcsdict.items():
    …
    for roleinx, dict_arg_arcs in sen_roleinx_argarcsdict.items():
        …
        for arg, list_arcs in dict_arg_arcs.items():
            for index_arcspostags in range(len(list_arcs)):
                arcs = list_arcs[index_arcspostags]
                postags = list_postags[index_arcspostags]
                #<省略：计算每个句子各类型句法的语义角色比例乘积>#
            # 计算贡献度
            for arc, list_arcratio in dict_arcsptgs_res.items():
                arcsptgs = arcsptgs + list_arcratio[0] / list_arcratio[1]
                count_arcspostags = count_arcspostags + 1
            arcsptgs = arcsptgs / count_arcspostags
            argmularcspostags = arcsptgs * self.ratio_normalized(argratio[1])
            if arg in dict_arg_res:
                temp = dict_arg_res[arg]
                temp[0] = temp[0] + argmularcspostags
                temp[1] = temp[1] + 1
                dict_arg_res.update({arg: temp})
            else:
                temp = [argmularcspostags, 1]
                dict_arg_res.update({arg: temp})
    for arg, list_argres in dict_arg_res.items():
        sen_res = sen_res + list_argres[0] / list_argres[1]
        count_sentence_result = count_sentence_result + 1
    if count_sentence_result != 0:
        sen_res = sen_res / count_sentence_result
    list_sentence_result.append(sen_res)
```

图 4.17: 语义贡献度计算核心代码

计算文书语义特征依赖句法分析和语义标注结果。计算出所有文书语义特征值后，3.2.2节给出了语义贡献度计算公式。图 4.17是计算文书平均语义贡献度的核心代码。

依据此分布，人工设定语义质量检测阈值为0.4972。经检测，达到阈值的裁判文书比率为92%。

4.3　Hadoop分布式集群

本节主要介绍了分布式集群架构的核心配置方法，包括HDFS HA 高可用配置和Hive、HBase的分布式配置。

4.3.1　HDFS HA高可用配置

平台采用HDFS的主从架构模型，集群由NameNode、SecondaryNameNode和DataNode组成，通常一个节点对应一台机器，管理数据存储。

(1) NameNode：负责文件系统命名空间管理、存储文件目录、block块和文件对应关系，以及block和DataNode的存储对应关系。

(2) SecondaryNameNode：是NameNode的冷备份，用来减少NameNode的工作量。

(3) DataNode：存储客户端Client发来的数据块，负责执行读写操作。

HDFS的一般架构中虽然存在SecondaryNameNode，但其无法提供“热备份”功能，一旦NameNode发生故障，系统需要停机。因此，平台采用HDFS 2.0的HA(High Availability) 架构，用于解决NameNode单点故障问题 [40]。

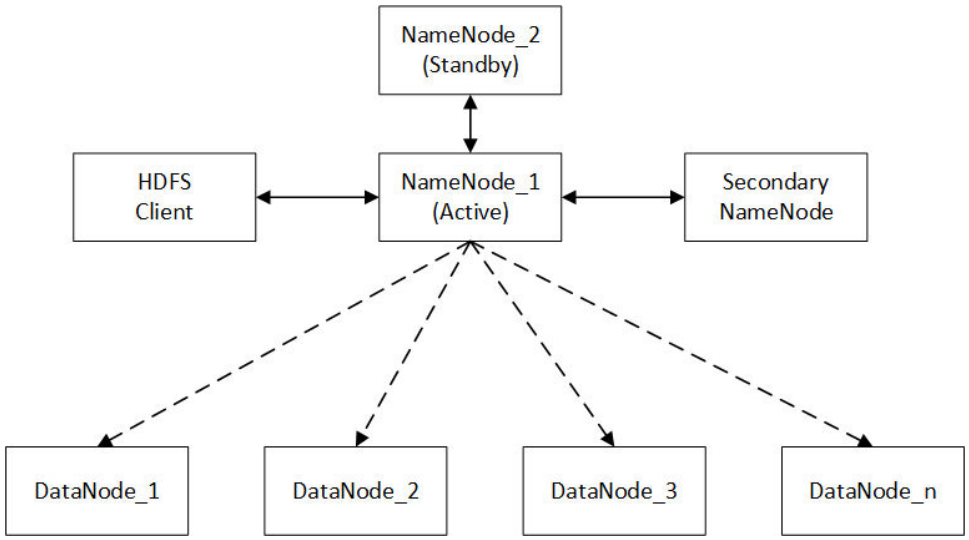

图 4.18: HDFS高可用架构

如图 4.18所示，平台拥有至少两个NameNode，一个处于Active状态，另一个处于Standby状态。Active NameNode对外提供服务，Standby NameNode仅同步Active NameNode的状态，以便能够在他失败时快速切换状态。

Hadoop使用ZooKeeper支持自动故障转移，ZooKeeper任务包括NameNode失败检测和NameNode选举。关键配置信息如图 4.20和图 4.19 所示。

```
<configuration>
    <property>
        <name>ha.zookeeper.quorum</name>
        <value>namenode01:2181,namenode02:2181,datanode01:2181</value>
    </property>
</configuration>
```

图 4.19: core.xml

```
<configuration>
    <!-- 访问代理类：client，mycluster，active 配置失败自动切换实现方式-->
    <property>
        <name>dfs.client.failover.proxy.provider.DQAnalysisPlatform </name>
        <value>org.apache.hadoop.hdfs.server.namenode.ha.ConfiguredFailoverProxyProvid
er</value>
    </property>
    <!-- 自动故障转移配置信息-->
<property>
    <name>dfs.ha.automatic-failover.enabled</name>
    <value>true</value>
    </property>
</configuration>
```

图 4.20: hdfs-site.xml

4.3.2　Hive、HBase分布式配置

Hive是基于Hadoop的数据仓库，支持标准SQL语法，本文将结构化数据存储在关系型数据库中。例如，文书时间信息和文书质量检测结果。Hive的元数据需要使用其他关系型数据库存储，平台配置了MySQL来存储元数据。Hive中的元数据包括表名、表的列和分区、表的属性以及表的所在目录等。Hive的关键配置信息如图 4.21 所示。

```
<property>
        <name>javax.jdo.option.ConnectionURL</name>
        <value>jdbc:mysql://localhost:3306/hahive</value>
</property>
```

图 4.21: hive-site.xml

HBase位于Hadoop生态底层，MapReduce为HBase提供高性能计算能力，HDFS为HBase提供存储支持。HBase主要配置信息如图 4.22所示，定义了HBase的存储目录和ZooKeeper的协调信息。

```
<configuration>
    <property>
        <name>hbase.rootdir</name>
        <!-- hbase 存放数据目录 -->
        <value>hdfs://namenode01:9000/opt/hbase/hbase_db</value>
    </property>
    <property>
        <name>hbase.cluster.distributed</name>
        <!-- 是否分布式部署 -->
        <value>true</value>
    </property>
    <property>
        <name>hbase.zookeeper.quorum</name> <!-- list of zookooper -->
        <value>namenode01,namenode02,datanode01,datanode02</value>
    </property>
    <property>
        <!--zookooper 配置、日志等的存储位置 -->
        <name>hbase.zookeeper.property.dataDir</name>
        <value>/opt/hbase/zookeeper</value>
    </property>
</configuration>
```

图 4.22: hbase-site.xml

4.4　平台数据交互

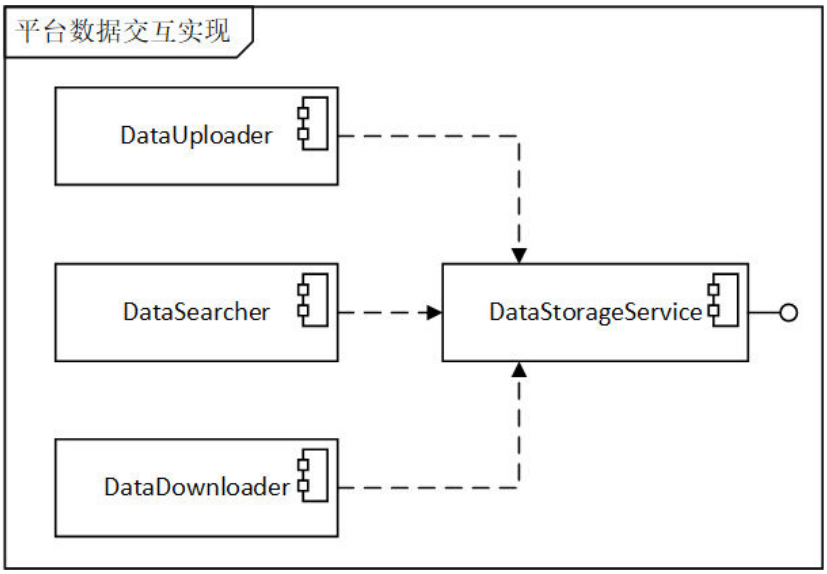

图 4.23: 平台数据交互实现

平台使用的文件操作类在Hadoop的org.apache.hadoop.fs包中，提供了数据

上传，数据检索和数据下载功能。图 4.23展示了平台数据交互的实现逻辑。

数据下载组件根据消息参数生成文件下载路径；扫描文件路径下所包含文件的状态，确认文件是否可以被下载；获取文件下载候选列表；执行下载操作。如果数据下载过程中发生异常，告警通知用户。图 4.24是下载示例，-p 是路径参数，按照列表中所有路径依次获取文书，以Json格式返回下载结果。

```
~ node01$ download -p '['/node03/hdfs/source/cases/453190.xml']'
File exist
Generating download json...
Success
```

图 4.24: 文书下载运行图

数据搜索组件根据消息中的文书案号，在给定集群中进行搜索；获取搜索结果并返回文件路径。若不存在则返回结果为空。图 4.25是搜索示例，-c 453190代表按照文书案号453190进行搜索，-n ‘[‘all’]’代表在集群所有节点搜索。

```
~ node01$ search -c 453190 -n '['all']'
Generating search info...
Success
```

图 4.25: 文书搜索运行图

图4.26以文书上传为例，给出了上传功能的主要代码。

```
public class DataStorageServiceImpl {
    public boolean upload(Path pathCaseDoc, Path pathCaseHdfs) throws IOException {
        //获取配置文件信息
        Configuration conf = new Configuration();
        conf.addResource(new Path("conf/core-site.xml"));
        //获取文件系统
        FileSystem hdfs = FileSystem.get(conf);
        //文件名称
        Path src = pathCaseDoc;
        Path dst = pathCaseHdfs;
        if(!hdfs.exists(dst)){
            hdfs.mkdirs(dst);
        }
        boolean tag = hdfs.copyFromLocalFile(src, dst);
        return tag;
    }
}
```

图 4.26: 文书上传核心代码

数据上传组件校验文书格式，对数据包解压缩后等待文书解析和质量检测结果，在数据质量合格的情况下，将原文本、解析和质量检测结果存储至数据库中。首先要获取Hadoop的core-site.xml配置文件信息，然后获取集群hdfs文件系统名称，最后调用函数上传。数据交互部分编译无误后生成JAR包，部署到Hadoop平台上，以接口方式提供数据交互服务。

4.5　平台用户权限管理

该模块面向平台管理员，管理用户权限。用户权限分为数据访问权限和指标设计权限。数据访问权限使用Apache Ranger插件实现，为HDFS、Hive和HBase提供了细粒度的权限管理接口；指标设计权限使用数据库验证手段实现，在用户向指标库中添加指标名时，权限验证通过方可添加。图 4.27表达了平台权限管理模块的实现逻辑。

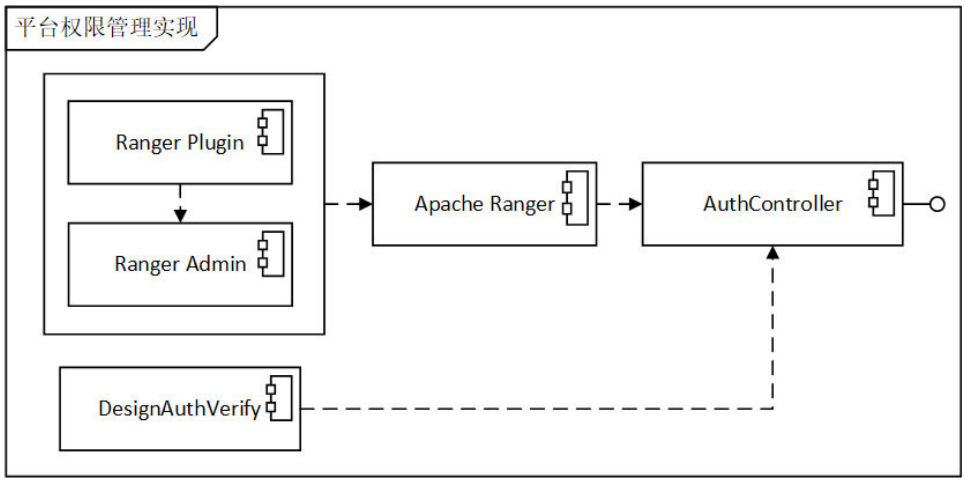

图 4.27: 平台权限管理实现

指标设计权限验证模块DesignAuthVerify根据用户请求数据解析用户id和开发机ip；在用户权限库中查询用户是否拥有权限。

Apache Ranger由Ranger Plugin和Ranger Admin构成。平台Hadoop 组件的数据访问控制由Ranger Plugin使用插件方式实现，Ranger Plugin定期轮询Admin获取权限控制策略。

根据3.6.1小节所述，RangerXxxxPlugin类是RangerBasePlugin 类的子类，其初始化过程由父类的初始化方法实现。该方法主要完成了以下几个功能：

(1) 调用cleanup()方法，清空refresher、serviceName、policyEngine变量值。

(2) 读取配置文件，并设置变量初始值：

- serviceType：Ranger访问控制服务类型。
- serviceName：Ranger访问控制服务名称。
- appId：Ranger提供服务的组件ID。
- propertyPrefix：Ranger插件的属性前缀。
- pollingIntervalMs：刷新器定期更新策略的轮询间隔时间。Ranger 插件会定期从Ranger Admin拉取新的策略信息，并保存在Hdfs缓存中。
- cacheDir：从Ranger Admin拉取策略到Hdfs插件的临时存放目录。

(3) 设置RangerAdminClient类的成员变量值。

(4) 调用createAdminClient()，创建RangerAdmin与RangerPlugin 通信的客户端，创建RangerAdminClient类的实例对象。

(5) 创建PolicyRefresher类的对象，调用startRefresher()开启策略刷新器，轮询定期从Ranger Admin获取更新的策略。

Ranger以插件方式向Hadoop的组件提供权限控制功能，并提供了可视化Ranger Web安全管理客户端。

以Hive为例，图 4.28给出了Ranger的用户配置信息，编辑install.properties文件，将RangerHivePlugin指向RangerAdmin。

```
POLICY_MGR_URL=http://namenode01.data.beta.cn0:6080
REPOSITORY_NAME=hivebeta
```

图 4.28: Ranger用户配置信息

启动插件后，出现图 4.29所示的界面，可以通过其管理用户的Hive数据访问权限。

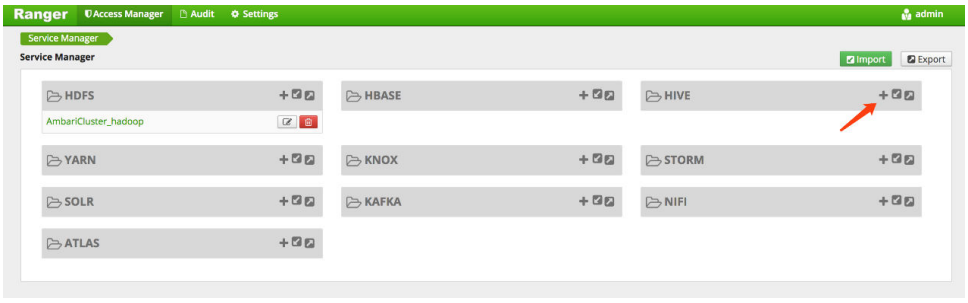

图 4.29: Hive权限管理界面

Ranger为Hive提供了细粒度的用户权限管理功能，可以对库、表和列进行授权。图 4.30是管理用户rangeruser1权限的界面截图。

图 4.30: Ranger细粒度用户授权

4.6　系统测试

系统测试是系统开发生命周期的重要环节。依据第三章的需求分析，本节对系统进行功能性测试和性能测试。通过设计测试用例的方式检验系统是否存在功能未实现、功能不完整或运行错误的情况。

4.6.1　测试环境

系统测试环境如表 4.6所示。集群计算机主节点内存大于8GB，从节点内存大于4GB。主节点安装Cloudera公司的Hadoop发行版组件管理客户端Cloudera Manager。主从节点均需要安装Cloudera CDH。平台所需的Hadoop组件都是通过Cloudera Manager统一管理和安装的。客户端需要正常运行Chrome浏览器的用户计算机。

表 4.6: 系统测试环境

设备名称	运行程序	角色说明
用户计算机 Mac OS 10.14	Chrome 72.0(64 位)	客户端，用于与平台交互
主节点 CentOS 7.4	Cloudera Manager 5.14 Cloudera CDH 5.14 Hadoop 2.9.2 Hive 2.0 HBase 2.0 MySQL 5.6	NameNode DataNode(HDFS集群角色) Resource Manager(YARN的核心角色) HMaster(HBase的角色) JobHistory Server(MapReduce的历史作业服务器角色) Hive Metastore Server(Hive的角色)
从节点 CentOS 7.4	Cloudera CDH 5.14 Hadoop 2.9.2 Hive 2.0 HBase 2.0	DataNode(HDFS集群角色) RegionServer(HBase角色) SecondaryNameNode(HDFS集群角色) NodeManager(YARN框架的角色)

4.6.2　功能测试

本小节的功能测试与设计章节的系统需求保持一致，从用户视角出发，模

拟用户的正常使用流程。对文书数据交互模块、文书解析模块、数据质量检测模块和平台用户权限管理模块进行功能测试。

如表 4.7所示，针对平台的数据交互功能进行测试。模拟用户操作流程，打包并上传文书后，平台自动进行质量检测服务，获取质量检测结果文件。结果表明，测试结果与预期结果相符。

表 4.7: 文书数据交互测试用例

用例编号	DQ_RS_01
测试目标	用户可正常上传文书，平台自动执行质量检测任务后，以Json格式返回检测结果
前置条件	用户拥有数据访问权限
正常流程	1. 用户将裁判文书以30个为一组打包为zip格式 2. 调用平台的数据上传接口，将文件上传 3. 平台解压缩文书数据包，自动执行文书解析和文书质量检测 4. 生成检测结果文件 5. 用户获取检测结果文件
预期结果	用户上传文书后，各功能模块正常调用，生成结果文件并返回。用户可以获取到检测结果文件
实际结果	与预期结果相符

表 4.8是针对平台文书解析接口的单元测试用例。使用MRUnit单元测试框架创建MapReduce的MapDriver对象，withInput指定文书路径，withOutput指定期望的文书解析结果，最后用runTest运行测试用例。测试结果符合预期。

表 4.8: 文书解析测试用例

用例编号	DQ_RS_02
测试目标	验证平台文书解析功能的有效性
接口函数	TextFieldParse类中的xmlFieldExtract函数
前置条件	文书压缩包上传至平台后解压成功
输入	在withInput中输入文书路径参数
预期输出	文书句法和语义解析的标注结果
实际结果	与预期相符

表 4.9是针对平台质量检测接口的单元测试用例。该用例同样使用MRUnit单元测试框架完成。测试结果符合预期。

表 4.9: 平台质量检测测试用例

用例编号	DQ_RS_03
测试目标	验证平台质量检测接口的功能有效性
接口函数	CalculationDim中的具体质量计算函数
前置条件	文书解析成功
输入	在withInput中以Text对象方式输入文书解析结果参数
预期输出	文书质量检测结果列表
实际结果	与预期相符

表 4.10是平台用户权限管理的测试用例。在该用例中，平台管理员可以新增用户并为用户设置平台组件的访问权限。授予HBase、Hive和HDFS的访问权限代表用户可以调用质量检测接口；关闭MySQL的权限代表用户无法使用质量指标设计接口添加质量检测指标。测试结果符合预期。

表 4.10: 平台用户权限管理测试用例

用例编号	DQ_RS_04
测试目标	平台管理员可新建用户并为用户设置数据访问权限，相关用户登录后只能在权限范围内调用接口
前置条件	用户拥有管理员权限
正常流程	1. 管理员登录Apache Ranger客户端 2. 管理员新建用户并为用户赋予HBase、Hive和HDFS的数据访问权限，关闭用户MySQL访问权限 3. 相关用户登录，调用文书质量检测接口 4. 调用文书质量指标设计接口，添加可调用指标名
预期结果	用户可以使用文书质检接口，但无法将设计的质量指标入库
实际结果	与预期结果相符

4.6.3　性能测试

文书质量检测平台的核心是质量指标计算，系统性能测试主要针对文书质量检测模块。用户对平台的访问请求是稀疏的，因此，本文重点关注忽略网络时延后的质量检测周期时长。

按照平台需求和设计章节描述，本文从文书质量检测章节的数据集中抽取文件大小最大的300篇裁判文书，设置了10组数据压缩包，测试平台对每个压缩

包的总计算时长是否在需求所述的90s 时间范围内。表 4.11是测试结果，实际测试结果符合预期。

表 4.11: 平台质量检测性能测试结果

样本	耗时	样本	耗时	需求耗时
$data_{01}$	40.10s	$data_{06}$	53.79s	≤90s
$data_{02}$	54.51s	$data_{07}$	47.28s	≤90s
$data_{03}$	37.29s	$data_{08}$	77.25s	≤90s
$data_{04}$	61.67s	$data_{09}$	42.01s	≤90s
$data_{05}$	43.09s	$data_{10}$	46.73s	≤90s

4.7　本章小结

本章介绍了裁判文书数据质量检测平台的实现细节。首先介绍了实现文书解析的具体方法，剖析了文书结构，给出了文书内容解析和语义解析的核心实现；然后介绍了文书质量检测的实现方法，并构造数据集进行实验，可视化展示了平台检测结果，设定了各指标的阈值，并以细致性指标和语义贡献度指标为例给出了文书内容质量检测和语义质量检测的核心代码；最后给出了Hadoop分布式集群的搭建中重要组件的配置方法，介绍了平台数据交互和权限管理的实现逻辑，以及对平台核心功能进行了系统测试。

第五章　总结与展望

5.1　总结

论文分析了法院信息化 3.0 背景下的司法数据质量问题，针对裁判文书数据提出了质量检测体系，分为文书结构化内容质量和文书非结构化语义质量。

- 文书内容质量模型结合了客观信息论和粗糙集的理论知识，构建了面向内容质量的九个指标，包括适配性、广阔性、细致性、遍及性、延迟性、持续性、包容性、丰富性和真实性。
- 文书语义质量使用自然语义处理中的方法，对非结构化的裁判文书进行句法分析和语义角色标注，提出了语义贡献度模型来度量文书语义质量。

同时，论文调研了司法数据质量在大数据背景下的挑战，明确了数据质量分析平台的系统边界。在此基础上，论文分析了平台需求，确定了平台基本功能，论述了平台工作原理，并在总体设计中给出了平台结构和功能模块划分。本文在系统详细设计部分，使用类图、顺序图等方式详细设计了平台权限管理模块、数据交互模块、内容解析模块和文书质量检测模块，并在模块设计后介绍了平台数据库设计。

在系统实现部分，描述了分布式平台的基本架构和关键配置信息，给出了文书内容解析和语义解析的核心代码，并以细致性指标和语义贡献度指标为例介绍了文书内容质量度量和语义质量检测的实现方式。以文书数据上传为例介绍了数据交互的实现，提供了Apache Ranger权限管理插件的使用方法。并使用真实数据实验，给出了实验结果，同时对系统进行了测试。

5.2　进一步工作

司法裁判文书质量检测平台目标是对审判执行流程中的文书类数据进行质量检测，目前该平台的功能较为单一，以接口方式面向开发者提供质检服务，其价值主要在于建设了较为完善的裁判文书质检体系。但从长远角度看，该体系不是一成不变的，平台也有诸多可优化之处：

- 文书语义质量检测精确性受限。因为其依赖于分词、词性标注、语义角色标注和依存句法分析的准确率。可是上述方法想达到高精确性需要构建司

法领域语料库，对相关算法模型进行训练。但是我们目前并没有完善的司法领域语料库，所以语义质量检测的精确性受限。后续将着手建立司法领域的语料库。

- 部分质量检测指标缺乏可解释性。部分指标使用了自然语言处理技术中的神经网络模型，该模型不具有可解释性。在质量检测领域，用户可能关注各计算环节的可解释性。而目前基于神经网络的机器学习和深度学习方法并不具备可解释性。因此，该问题有两个潜在的研究方向，一是攻克神经网络模型的可解释性问题，二是重新构建具有可解释性的指标。

- 与已有司法业务数据库的对接。目前我国文书型数据以字段的形式保存在关系型数据库中，想要将其转移至大数据平台的数据库中，需要设计完备的大数据平台表结构。目前已有表结构较为简单，需要根据业务拓展。

- 与已有司法应用的对接。目前平台仅以接口形式向开发者提供质量检测服务，依赖于已有司法应用。与旧的司法应用对接存在一定的复杂度，所以这部分工作一直处于延后状态，后续系统将积极适配旧的司法应用。

参考文献

[1] 孙海龙, 高伟, 裁判文书及其公信力现状调查和改革路径研究, 法律适用(2007) 35–40.

[2] 范明志, 网络司法公开："互联网+司法"改革的起跑线, 人民论坛(2018) 92–94.

[3] 中国裁判文书网, http://wenshu.court.gov.cn (2019).

[4] T. He, H. Lian, Z. Qin, Z. Zou, B. Luo, Word embedding based document similarity for the inferring of penalty, in: Web Information Systems and Applications - 15th International Conference, WISA 2018, Taiyuan, China, September 14-15, 2018, Proceedings, 2018, pp. 240–251.

[5] T. He, H. Lian, Z. Qin, Z. Chen, B. Luo, PTM: A topic model for the inferring of the penalty, J. Comput. Sci. Technol. 33 (4) (2018) 756–767.

[6] R. Y. Wang, D. M. Strong, Beyond accuracy: What data quality means to data consumers, J. of Management Information Systems 12 (4) (1996) 5–33.

[7] C. Batini, M. Palmonari, G. Viscusi, The many faces of information and their impact on information quality, in: Proceedings of the 17th International Conference on Information Quality, IQ 2012, Paris, France, November 16-17, 2012., 2012, pp. 212–228.

[8] T. Saaty, Decision making with the analytic hierarchy process, International Journal of Services Sciences 1 (2008) 83–98.

[9] S. Feng, L. D. Xu, Decision support for fuzzy comprehensive evaluation of urban development, Fuzzy Sets and Systems 105 (1) (1999) 1–12.

[10] 李德毅, 刘常昱, 论正态云模型的普适性, 中国工程科学6 (8) (2004) 28–34.

[11] D. McGilvray, Executing data quality projects, Morgan Kaufmann, San Francisco, 2008, pp. 303 – 306.

[12] 许建峰, 汤俊, 马雪峰, 徐斌, 沈艳丽, 乔永杰, 客观信息的模型和度量研究, 中国科学:信息科学45 (3) (2015) 336–353.

[13] J. Yao, Y. Zhang, A scientometrics study of rough sets in three decades, in: Rough Sets and Knowledge Technology - 8th International Conference, RSKT 2013, Halifax, NS, Canada, October 11-14, 2013, Proceedings, 2013, pp. 28–40.

[14] Z. Li, M. Zhang, W. Che, T. Liu, W. Chen, H. Li, Joint models for chinese POS tagging and dependency parsing, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL, 2011, pp. 1180–1191.

[15] W. Che, Z. Li, Y. Li, Y. Guo, B. Qin, T. Liu, Multilingual dependency-based syntactic and semantic parsing, in: Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2009, Boulder, Colorado, USA, June 4, 2009, 2009, pp. 49–54.

[16] H. Lian, T. He, Z. Qin, H. Li, J. Liu, Research on the information quality measurement of judicial documents, in: 2018 IEEE International Conference on Software Quality, Reliability and Security Companion, QRS Companion 2018, Lisbon, Portugal, July 16-20, 2018, 2018, pp. 178–181.

[17] C. Batini, C. Cappiello, C. Francalanci, A. Maurino, Methodologies for data quality assessment and improvement, ACM Comput. Surv. 41 (3) (2009) 16:1–16:52.

[18] J. Hu, K. Huang, K. Kuse, G. Su, K. Wang, Customer information quality and knowledge management: A case study using knowledge cockpit, J. Knowledge Management 1 (3) (1997) 225–236.

[19] F. Naumann, C. Rolker, Assessment methods for information quality criteria, in: Fifth Conference on Information Quality (IQ 2000), 2000, pp. 148–162.

[20] S. Schelter, D. Lange, P. Schmidt, M. Celikel, F. Bießmann, A. Grafberger, Automating large-scale data quality verification, PVLDB 11 (12) (2018) 1781–1794.

[21] D. Firmani, M. Mecella, M. Scannapieco, C. Batini, On the meaningfulness of "big data quality" (invited paper), Data Science and Engineering 1 (1) (2016) 6–20.

[22] 蔡莉, 梁宇, 朱扬勇, 何婧, 数据质量的历史沿革和发展趋势, 计算机科学45 (4) (2018) 1–10.

[23] G. R. Klare, Assessing readability, Reading Research Quarterly 10 (1) (1974) 62–102.

[24] J. B. Miner, Review of the mind of the buyer, a psychology of selling, Journal of Applied Psychology 6 (1922) 215.

[25] C. Batini, M. Scannapieco, Data and information quality - dimensions, principles and techniques, Data-Centric Systems and Applications, Springer, 2016.

[26] C. Kiefer, Assessing the quality of unstructured data: An initial overview, in: Proceedings of the Conference "Lernen, Wissen, Daten, Analysen", Potsdam, Germany, September 12-14, 2016., 2016, pp. 62–73.

[27] 中国裁判文书网, FYB/T 51301.1-2018 审判信息数据质量要求第1部分：结构化案件信息(2018).

[28] M. Zhang, Z. Deng, W. Che, T. Liu, Combining statistical model and dictionary for domain adaption of chinese word segmentation, Journal of Chinese Information Processing 26 (2) (2012) 8–12.

[29] D. Shao, N. Zheng, Z. Yang, Z. Chen, Y. Xiang, Y. Xian, Z. Yu, Domain-specific chinese word segmentation based on bi-directional long-short term memory model, IEEE Access 7 (2019) 12993–13002.

[30] X. Sun, H. Wang, W. Li, Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection, in: The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers, 2012, pp. 253–262.

[31] X. Zheng, H. Chen, T. Xu, Deep learning for chinese word segmentation and POS tagging, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, 2013, pp. 647–657.

[32] 中文语义角色标注的特征工程, 中文信息学报21 (1) (2007) 79–84.

[33] W. Che, Z. Li, T. Liu, LTP: A chinese language technology platform, in: COLING 2010, 23rd International Conference on Computational Linguistics, Demonstrations Volume, 23-27 August 2010, Beijing, China, 2010, pp. 13–16.

[34] N. Xue, Labeling chinese predicates with semantic roles, Computational Linguistics 34 (2) (2008) 225–255.

[35] N. Xue, F. Xia, F. Chiou, M. Palmer, The penn chinese treebank: Phrase structure annotation of a large corpus, Natural Language Engineering 11 (2) (2005) 207–238.

[36] M. Palmer, P. Kingsbury, D. Gildea, The proposition bank: An annotated corpus of semantic roles, Computational Linguistics 31 (1) (2005) 71–106.

[37] T. White, Hadoop: The definitive guide, Southeast University Press, 2011.

[38] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113.

[39] 傅郁林, 民事裁判文书的功能与风格, 中国社会科学1 (4) (2000) 123–133.

[40] Y. Kim, T. Araragi, J. Nakamura, T. Masuzawa, A distributed and cooperative namenode cluster for a highly-available hadoop distributed file system, IEICE Transactions 98-D (4) (2015) 835–851.

简历与科研成果

基本情况 廉昊，男，汉族，1995 年 2 月出生，山西省晋中市人。研究生期间随陈振宇、何铁科老师参与国家重点研发计划项目“多元智能化诉讼服务及审判执行关键技术研究”，承担课题5“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”中关于司法数据质量和类案推荐的部分研究工作。

教育背景

2017.9～2019.6 南京大学软件学院 硕士

2013.9～2017.7 山西大学计算机与信息技术学院 本科

论文

1. He, T., **Lian, H.**, Qin, Z., Chen, Z., Luo, B., 2018. PTM: A topic model for the inferring of the penalty. J. Comput. Sci. Technol. 33, 756 – 767.
2. He, T., **Lian, H.**, Qin, Z., Zou, Z., Luo, B., 2018. Word embedding based document similarity for the inferring of penalty, in: Web Information Systems and Applications-15th International Conference, WISA 2018, Taiyuan, China, September 14-15, 2018, Proceedings, pp. 240 – 251.
3. **Lian, H.**, Qin, Z., Song, H., He, T., 2018. E-CAT: evaluating crowdsourced android testing, in: Data Science - 4th International Conference of Pioneering Computer Scientists, Engineers and Educators, ICPCSEE 2018, Zhengzhou, China, September 21-23, 2018, Proceedings, Part I, pp. 493 – 504.
4. Qin, Z., He, T., **Lian, H.**, Tian, Y., Liu, J., 2018. Research on judicial data standard, in: 2018 IEEE International Conference on Software Quality, Reliability and Security Companion, QRSCompanion2018, Lisbon, Portugal, July16-20, 2018, pp.175 – 177.
5. **Lian, H.**, He, T., Qin, Z., Li, H., Liu, J., 2018. Research on the information quality measurement of judicial documents, in: 2018 IEEE International Conference

on Software Quality, Reliability and Security Companion, QRS Companion 2018, Lisbon, Portugal, July 16-20, 2018, pp. 178 – 181.

6. Qin, Z., **Lian, H.**, He, T., Luo, B., 2017. Cluster correction on polysemy and synonymy, in: 14th Web Information Systems and Applications Conference, WISA2017, Liuzhou, Guangxi Province, China, November 11-12, 2017, pp. 136 – 138.

7. **Lian, H.**, Qin, Z., He, T., Luo, B., 2017. Knowledge graph construction based on judicial data with social media, in: 14th Web Information Systems and Applications Conference, WISA2017, Liuzhou, GuangxiProvince, China, November11-12, 2017, pp. 225 – 227.

8. Shen, S., **Lian, H.**, He, T., Chen, Z., 2017. Clustering on the stream of crowdsourced testing, in: 14th Web Information Systems and Applications Conference, WISA2017, Liuzhou, Guangxi Province, China, November 11-12, 2017, pp. 317 – 322.

专利

1. 何铁科，**廉昊**，严格，陈振宇，李玉莹，“ 一种基于词嵌入的协同过滤推荐方法”，申请号：201810561270.8，已受理。

2. 何铁科，严格，**廉昊**，秦泽民，史洋洋，陈振宇，“一种用于刑罚推断的主题模型PTM”，申请号：201810561189.X，已受理。

3. 何铁科，**廉昊**，严格，陈振宇，李玉莹，“ 一种规则约束下的文本信息质量度量方法”，申请号：201810561187.0，已受理。

4. 陈振宇，何铁科，**廉昊**，秦泽民，骆斌，李玉莹，张欣，“一种问答社区中的感知信息质量度量方法”，申请号：201711200605.5，已受理。

5. 陈振宇，何铁科，秦泽民，**廉昊**，骆斌，李玉莹，张欣，“一种基于k-means的类案推荐方法”，申请号：201711200604.0，已受理。

致　　谢

在本篇论文完成之际，我要向所有关心帮助过我的人致以最诚挚的感谢。

首先感谢我的研究生导师何铁科老师，他在整个研究生阶段给予我很大帮助，让我认识到作为一名学者应对学术抱有严谨和勤奋的态度，对生活持有热情和乐观的心境，同时身体是革命的本钱，无论多忙都要强身健体。在论文写作时导师对论文的不足之处做出指正，并提出了很多修改和优化的建议。何老师渊博的学识和严谨的治学态度，我会一直谨记于心。

同时，我要感谢在腾讯实习期间给予我帮助的所有同事们，谢谢诸位在技术上对我的悉心指导，让我接触到了业界最新的技术，并在遇到困难的时候能够耐心的分析并解决问题。这是一段于我而言非常宝贵的实践历程。

感谢南京大学软件学院智能软件工程(Intelligent Software Engineering, iSE)实验室，感谢陈振宇老师的悉心指导。开放的教学环境和先进的教学思想帮助我们快速的成长。这两年的硕士时光是我进步最大的两年，让我接触了庞大的项目，长了很多见识。认真努力的同学和学识渊博的老师也让我对自身提出了更高的要求。更宽广的业界视野，更合理的未来规划让我充满信心面对未来。

谢谢我的家人，在生活上给予了我莫大的帮助，让我可以专心于学业，同时感谢我的女朋友，在我写作遇到困难时伸出援手。

最后感谢为本文答辩评审的各位老师们，谢谢诸位的辛勤工作。

版权及论文原创性说明

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：　　年　月　日