南京大學

研 究 生 毕 业 论 文

（申请工程硕士学位）

论 文 题 目	数据质量驱动的司法文本扩增系统设计与实现
作 者 姓 名	许子桓
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

2021 年 5 月 18 日

学　　　　号　：MF1932215

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 Judicial Text Augmentation System

By

Zihuan Xu

Supervised by

Professor **Zhenyu Chen**

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 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：数据质量驱动的司法文本扩增系统设计与实现

工程硕士（软件工程领域）专业 2019 级硕士生姓名：许子桓

指导教师（姓名、职称）：陈振宇 教授

摘 要

伴随信息技术在司法行业的广泛应用，越来越多的司法业务数据迁移到了信息系统中。在最高人民法院的统筹领导下，我国顺利建成裁判文书网、审判流程信息公开网等司法平台，一定程度上完成了司法业务的信息化。司法信息化带动了司法智慧化的发展，司法部提出“数字法治、智慧司法”的指导意见。当前，代表性的智慧司法应用包括类案自动推荐、法条自动预测和量刑自动预测等。这类应用的基础都是深度学习模型，深度学习模型需要大量的司法文本作为输入的训练数据。当前智慧司法应用的落地存在一个问题，就是司法文本数量不够且质量不高，这就导致训练出来的模型应用效果不好、且泛化能力差。因此获取足够数量以及较高质量的司法文本，成为了司法技术研究者的需求痛点。

本文设计并实现了一个数据质量驱动的司法文本扩增系统，相比于传统的数据扩增方法，本系统聚焦于司法领域，从数据质量的角度出发实现扩增。针对司法裁判文书这类半结构化数据，文本质量可以在司法领域特征内划分为客观和主观两个维度，然后基于这两个维度设计扩增方法。对于客观质量维度的扩增，取裁判文书中具备结构化特征的部分，结合文本数据扩增基本规则和客观信息论，从归纳的质量维度对司法文本进行扩增，本文共设计了五种客观维度的扩增方法，具体包括延迟性扩增、真实性扩增、完整性扩增、一致性扩增和易读性扩增，五种方法分别从司法文本的五个客观质量维度进行扩增。对于主观质量维度的扩增，选取裁判文书中结构化无法体现的文本质量维度，本文聚焦文本中包含的语义信息，模拟人的主观判断，结合机器学习技术，用 word2vec 学习司法文本的特征，实现司法语言特征扩增，并结合情感分析技术实现情感中立性扩增。

本文技术的实现以 Python 的 web 框架 Django 为基础，将扩增后的司法裁判文书以文本文件的格式输出。对于文件管理，采用 HDFS 文件系统存储文件数据，这种分布式存储服务可以有效提高文件存储的可拓展性。经实验验证，本文设计的司法文本扩增系统能够扩增出高质量司法文本，继而用以提高智慧司法机器学习模型的质量。

关键词：司法文本，数据扩增，数据质量，自然语言处理

南京大学研究生毕业论文英文摘要首页用纸

THESIS: The Design and Implementation of Data Quality-Driven Judicial Text Augmentation System

SPECIALIZATION: Software Engineering

POSTGRADUATE: Zihuan Xu

MENTOR: Professor Zhenyu Chen

Abstract

With the widespread application of information technology in the judicial industry, more and more judicial data has been migrated to information systems. Under the guidance of the Supreme People's Court of The People's Republic of China, China has successfully established Judgements Online, Judicial Process Information Online and other judicial platforms, which achieved the informatization of judicial business to a certain extent. Judicial informatization has promoted the development of judicial wisdom. The Ministry of Justice of the People's Republic of China put forward the instruction of "digital rule of law and intelligent justice". Currently, representative intelligent judicial applications include automatic recommendation of similar cases, automatic prediction of legal provisions and automatic prediction of sentencing. The foundation of these applications is deep learning model, which requires a large amount of judicial text as training data. There is currently a problem in the implementation of intelligent judicial applications, that is, the amount of judicial text is insufficient, while the quality is not high, which leads to poor generalization ability of the models and poor application effect. Therefore, obtaining sufficient quantity and high-quality judicial text has become a demand of judicial technology researchers.

This thesis designs and implements a system for augmenting high-quality judicial text. Compared with the traditional data augmentation method, this system focuses on the judicial field and implements the data augmentation from the perspective of data quality. For semi-structured data such as judicial documents, the data quality can be divided into objective and subjective dimensions within the characteristics of the judicial field. Then the data augmentation method is designed based on the two dimensions.

For the augmentation of objective quality dimension, take the parts of judicial documents that have structural features, combine the basic rules of text data augmentation and objective information theory, and augment judicial texts from the summarized data quality dimensions. The thesis proposes five augmentation methods in objective quality dimension totally, including delay augmentation, authenticity augmentation, integrity augmentation, consistency augmentation and readability augmentation. The five augmentation methods are related to five objective quality dimensions respectively. For the augmentation of subjective quality dimension, take the data quality dimensions that cannot be reflected in the structure of judicial documents. This thesis focuses on the semantic information contained in the texts, simulates human subjective judgments, combines machine learning technology, learns the features of judicial texts with word2vec, and then implements judicial language features augmentation. Furthermore, the thesis implements emotional neutrality augmentation combined with sentiment analysis technology.

The implementation of the technology in this thesis is based on Django, which is a Python web framework. The augmented judicial documents are output in the format of a text file. For file management, the HDFS file system is used to store data. The distributed storage service can effectively improve the scalability of file storage. Experiments have shown that the judicial text augmentation system designed in this thesis can augment high-quality judicial texts, which can then be used to improve the quality of smart judicial machine learning models.

Keywords: Judicial Text, Data Augmentation, Data Quality, Natural Language Processing

目录

表 目 录

插图

第一章 引言

1.1 选题背景与研究意义

随着人工智能技术逐步渗透到各个领域，很多行业的工作方式得到了改进，工作效率也得到了提高。近些年，国内外司法机关以及相关研究机构开始将人工智能引入到司法业务中 [1]。2016 年，在我国最高人民法院信息化建设工作领导小组召开的重要会议上，周强组长提出了我国要建成智能化、自动化智慧法院的提案。当前，我国也正在推进形成大平台共享、大系统共治、大数据慧智的信息化格局。智慧法院要求司法部门对传统司法业务进行技术变革，并融合人工智能技术，以数字化、自动化的方式辅助法律咨询、司法判决和文书管理等司法业务。在智慧法院的背景下，我国司法智能研究人员提出了一系列基于人工智能的司法应用，包括多元纠纷结果预测、类案自动推荐 [2] 和罪名自动预测 [3] 等。

司法行业的人工智能应用大多是筛选预测类应用，这类应用的通用技术方法是学习大数据集的数据特征，然后根据学习训练得到的模型进行司法案例的筛选预测。模型预测的准确率受两个因素影响，一是选择的机器学习算法，二是使用的训练数据集。机器学习算法决定了数据的处理方法和模型网络的构造结构，从训练过程的角度影响着最终模型，而训练数据集则从根源影响着最终模型，如果数据集质量差，则训练结果也会不尽如人意。所以获取训练数据集是非常重要的一步。在司法领域，由于各司法机构和机关单位长期以来没有统一的数据存储格式，所以导致了司法数据来源复杂，且结构各异。这也间接导致了在训练机器学习模型时，缺少适合作为训练集的高质量数据。缺少数据会使得机器学习模型泛化能力差，场景覆盖能力低，甚至功能残缺。如果用质量参差不齐的数据进行训练，则会使预测结果准确率低，且模型偏差不可预测。因此，即使已经有一些不错的机器学习预测算法被引入到司法领域，但当前司法智能应用大多都还处于技术研究阶段，真正投入工业界使用的还很少。如何获取大量高质量的司法文本成为了至关重要的问题。

本文针对上述问题，基于机器学习技术、情感分析技术和客观信息论，设计并实现了一个数据质量驱动的司法文本扩增系统。该系统的核心功能是：基于给出的司法文本，扩增出更多高质量司法文本。系统的数据扩增模块使用的是本文专门设计的司法文本扩增技术，该技术将司法文本的数据质量拆分为主观

和客观两个维度，然后设计了主观质量维度的扩增方法和客观质量维度的扩增方法。一篇高质量司法文本经过主客观两个维度的扩增，可以得到更多新的高质量司法文本。通过这种方法，能够为司法智能的机器学习模型提供更高质量的训练数据集。

1.2　国内外研究现状

司法数据扩增的对象是司法文本，本文针对司法文本设计了主观质量维度的扩增方法和客观质量维度的扩增方法，这两种方法都是基于司法文书数据质量设计出来的。因此本部分调研了文本类型数据扩增研究现状和司法领域文本质量研究现状。

1.2.1　文本类型数据扩增研究现状

对于机器学习模型的训练来说，如果缺少足够数量的训练数据，或者训练数据的分布较为集中，则会导致训练出来的深度神经网络泛化能力较差，模型很难覆盖所有可能的情况。在缺少更多类别训练数据的情况下，通常会往训练数据集中增加同类数据，但是如果训练集的数据相似度过高，则会导致学习得到的神经网络过拟合，从而降低模型预测的准确率。对于过拟合的情况，有 Dropout [4] 和 Batch Normalization [5] 等方法对训练数据集进行处理，但这些方法都是在训练数据有明显缺陷的情况下进行后期补救，最根本解决训练数据数量不足、质量不高的方法就是数据扩增。

数据扩增是基于已有数据，对其进行变换操作，得到数据含义不变的新数据的过程。数据扩增能够将一份数据变成两份或更多数据，以此来解决机器学习训练集数据不足的问题。数据扩增不仅可以增加数据的数量，也可以丰富数据的种类，例如某类数据的数量少，可以基于已有的此类数据，扩增出更多的同类数据，这样不仅保证了数据集的覆盖全面性，还可以提高机器学习模型的泛化能力。根据扩增对象的不同，数据扩增可以分为图像扩增和非图像扩增，非图像扩增又可以分为结构化数据扩增和文本数据扩增。图像扩增是数据扩增的主要研究领域，图像的扩增可以通过图像旋转、翻转、裁剪等方式实现，近些年，也有通过对抗网络和变分编码器等技术来实现的。非图像扩增中的结构化数据扩增主要是针对可以用 MySQL 存储的数字数据，用设计好的数学公式对数字进行计算，从而实现扩增。非图像扩增中的文本扩增和前两种扩增完全不一样，图像扩增是对连续的像素点进行数据变换实现扩增，它的数据具有连续性，而文本由汉字组成，一个个汉字是离散的数据，因此无法用图像扩增方法对文本进行处理，同样也无法用数学公式直接进行文本数据的扩增。因此，文本扩增相

较于其他数据的扩增，难度要更大 [6]。现有的文本扩增方式有文本裁剪、同义词替换、回译、句法变换和基于机器学习的扩增。文本裁剪随机选择文本的某个部分，将其删除，得到缺失部分词语的新文本；同义词替换将文本中的词汇替换成词义相近的词语；回译先将原文本翻译成其他语言，然后再翻译回中文，得到新文本；句法变换包含缩句和扩句两种方式，缩句提取原文本语句的主要结构，组合成新句，扩句向原文本添加状语、定语等辅助词汇实现语句扩充；基于机器学习的扩增先训练大量司法文本，学习文本内部特征，得到机器学习模型，通过模型实现对文本的扩增。

上述文本扩增方法是面向所有领域的，属于通用方法，对于司法领域的扩增，当然也可以使用这些方法。但是司法文本也具备领域特殊性，例如司法文书结构、司法专业用语等，这些是其他领域文本不具备的。针对司法文本的特殊性，结合通用领域的扩增方法，Guo 等人提出了 TauJud 框架 [7] 用于司法文书的扩增，Zhang 设计了司法文本自动化生成方案 [8]。TauJud 先后用停用词删除、回译、文档裁剪、Blind 扩增、反事实扩增和同义词替换来实现司法文本的扩增，缺点是只针对司法文本中涉案人员、案发地点等少数字段进行了司法领域定制扩增。司法文本自动化生成方案基于 TauJud 设计了扩增方法，但是并没有特别关注数据质量，缺乏对数据质量的保证。

基于上述司法领域数据扩增研究现状，本文从司法数据质量的视角出发，结合已有司法文本扩增技术，并加入机器学习词向量技术，扩增出质量更高的司法文本，用于相关司法推荐模型的训练。

1.2.2　司法领域文本质量研究现状

司法文本的数据质量对司法智能机器学习模型的质量有很大的影响。对司法文本的质量研究也是司法智能非常重要的一部分。在我国建成裁判文书公开平台之前，对文书质量的研究还是集中于规范性上，研究主体也仅限于司法从业人员。Hu [9] 指出说理是司法民主和理性的体现，并认为我国的裁判文书存在说理不当、千篇一律的问题。Zhang 等人 [10] 对我国裁判文书的撰写提出了规范格式、繁简得当的要求，也说明了案情描述清晰的重要性。Han [11] 构建了三段论式的文书说理模式，对我国裁判文书的撰写具有重要的指导意义。

早期司法文本质量的研究主要是为了指导裁判文书的撰写，很少有学者提出具体的度量方法。Zhang [12] 提出了语言文字、案件事实、裁判理由和裁判结果四个静态指标用于评价司法文本。Yang [13] 将三段论式说理与司法业务相结合，提出了司法文本的质量度量指标。两种方法虽然给出了文书质量的评价指标，但是具体的评价操作都还是基于人工评分，很难避免人的主观情感对评价

结果的影响，也很难保证大数据背景下的质量评估效率。

信息化背景下的司法文本质量度量方法参考了过往的裁判文书质量度量指标，将基本的文书结构评价与深层次的语义评价相结合，得到了一些质量评价的方法。Li [14] 基于 XG-Boost 和深度残差网络设计了文书说理全面性和法条引用准确性的度量方法。Lian [15] 基于粗糙集和信息论，并结合司法文本的特征，提出了裁判文书质量度量指标，具体指标信息如表 1.1所示。

表 1.1: 裁判文书语义质量度量指标

语义指标	**度量方法**
谓词角色	语义谓词在语法树中的角色
谓词词性	谓词在句法树中的词性
谓词数目	语义谓词的数量
语义角色	语义角色的比例
语义角色范围内句法树角色	语义角色范围内句法树角色比例
语义角色范围内的词性	语义角色范围内单词的句法树角色比例
语义角色类内句法树角色	语义角色类内句法角色占比
语义角色类内的词性	语义角色范围内单词的词性角色占比

Li [16] 在 Lian 提出的裁判文书质量度量指标的基础上，设计了裁判文书的客观和主观两个角度的度量指标，并给出了指标的计算公式，其中客观指标针对文书的规范性进行表层质量的度量，主观指标则对文书深层信息进行质量度量。

1.3 本文主要研究工作

为了保障司法智能机器学习应用有足够的高质量训练数据，本文设计了一种数据质量驱动的司法文本扩增技术，并基于该技术，开发了一个可用的数据扩增系统。数据质量驱动的司法文本扩增分为主观质量维度的扩增和客观质量维度的扩增，主观质量维度的扩增旨在确保扩增出语义明确、符合人思维方式的高质量文书，客观质量维度的扩增旨在确保扩增出逻辑清晰、符合司法文书格式的高质量文书。两类扩增方法相结合，就可以扩增出符合机器学习算法所需的高质量司法文本数据。

本文的工作分为以下两部分: 数据质量驱动的司法文本扩增技术的设计、司法文本数据扩增系统的实现。

对于数据质量驱动的司法文本扩增技术的设计，本文借鉴了已有的文本扩增方法，并结合司法领域的特殊性，总结出高质量司法文本所具备的性质，然后

设计出针对司法裁判文书的数据扩增技术。针对司法文本语言的特殊性，本文用 Word2vec 机器学习方法学习高质量司法文本的语言特征，然后用得到的 CBOW 模型扩增出具备司法语言特征的裁判文书；针对司法文本的情感中立性，本文将现有的情感词典辅以机器学习算法，融合得到一个司法领域的情感词库，利用词库中词语的情感标签，确保扩增的文本在情感上是中立的；针对司法文本的结构逻辑性，本文设计了延迟性扩增、真实性扩增、完整性扩增、一致性扩增和易读性扩增总共五种扩增方法，对文本的相关内容字段进行扩增，保证了扩增后的文本在相关时间节点上是合理的、对案件的还原是真实的、在结构上是完整的、前后语言规范是一致的，以及词语句式是简明的。

对于司法文本数据扩增系统的实现，本文构建了由前端、服务端和存储介质共同组成的 web 系统。前端负责展示系统界面，与用户友好地交互。用户发送的请求会通过 Https 协议传输到系统的服务端，服务端响应并处理该请求，然后把处理结果返回至前端，由前端展示在 UI 界面。服务端由文书解析模块、主观质量维度的扩增模块和客观质量维度的扩增模块组成。文书解析模块从存储设备中获取用户上传的待扩增文本，然后对其进行格式转换、分词和标签提取操作，将文书解析成扩增模块可处理的格式。处理后的标准 json 格式文件被传递到扩增流程中，完成主观和客观两个维度的扩增。扩增系统的存储介质是 MySQL 数据库和 HDFS。所有结构化数据会被存储在 MySQL 中，包括了用户名、登录密码、操作权限、用户文件名和系统文件名等。用户上传的原文件以及扩增后的文本会被存储在 HDFS 中。

1.4　本文的组织结构

本文共有五章，各章的主要内容和组织逻辑如下：

第一章引言，介绍了智慧司法背景下，司法文本扩增的意义，调研了国内外文本类型数据扩增和司法领域文本质量的研究现状，简要说明了本文的主要工作。

第二章技术综述，讲解了本文设计的数据扩增系统所用到的一系列技术，包括自然语言处理技术、数据扩增技术、数据质量评估技术、Django 和分布式存储技术。其中自然语言处理技术又分为了中文分词技术、词向量技术和情感分析技术，数据扩增技术分为了通用数据扩增技术和司法文本数据扩增技术。

第三章需求分析与概要设计，先简述了系统的整体架构，接着分为功能性需求和非功能性需求两个方面，详细分析了系统的需求。然后结合 4+1 视图，介绍了系统的总体设计，并根据总体设计详细阐释了数据交互模块、数据管理模块、文书解析模块和数据扩增模块的设计。

第四章系统实现与实验测试，基于上一章的需求分析与技术设计，对数据交互模块、数据管理模块、文书解析模块和数据扩增模块的实现进行了详尽的说明。然后对设计的扩增技术进行了实验分析。实验之后测试了系统的功能点，并完成了性能测试。实验和测试的结果共同证明了该技术的应用价值和系统的可用性。

第五章总结与展望，对论文的工作进行了总结，并指出了本文设计的扩增技术以及对应系统的不足之处，针对这些不足给出了未来工作的改进方向。

第二章　技术综述

2.1 自然语言处理技术

本文所设计的司法文本扩增系统需要对中文文本进行处理，这个过程用到了自然语言处理技术，其中包括了分词技术、词向量技术和情感分析等。分词技术主要用于文本语句的分割，词向量技术主要用于从语义的角度量化词语之间的相似度，情感分析主要用于保障文本语句的情感中立性。

2.1.1 中文分词技术

分词的任务是要把一段连续的文字分割成词语序列。由于中英文的词句构成不同，所以两者的分词过程有很大的区别。英文单词之间有空格分开，因此对于词语的识别相对简单。而中文单词之间并没有任何分隔符，所以分割出语义正确的词语需要经过更复杂的文本分析工作。由此可以得知，中文分词的技术实现难度要远远高于英文分词。

分词是大多数自然语言处理任务的必备步骤。包括本文提出的技术在内，很多自然语言处理应用的最上游都是分词，因此分词技术是数据扩增中极其重要的一环。当前常见的中文分词技术有三个分支，分别是基于字符串匹配的分词方法、基于理解的分词方法和基于统计的分词方法 [17]。

基于字符串匹配的分词方法会遵循某些特定的规则，扫描某个机器词典，查询词典中是否存在和指定字符串完全相同的词条，一旦查询到词条就不再继续扫描。所以它也被称为基于词典的分词方法。如果扫描方向是从前往后，则属于正向匹配法，如果扫描的方向是从后往前，则属于逆向匹配法；如果优先选择长词汇，属于最长匹配法，如果优先选择短词汇，则属于最短匹配法；如果不需要提前标注词性，属于简单分词法，如果需要提前标注词性，则属于词性标注分词法。综合上述方法，研究人员总结出分词表现较为优异的四种匹配，包括了从左到右的正向最大匹配、从右到左的逆向最大匹配、切分词数最少的最小切分匹配和双向扫描的双向最大匹配 [18]。字符串匹配法相比其他方法速度更快，时间复杂度为 $O(n)$ ，并且实现简单，缺点是无法识别歧义词和未登录词。

基于理解的分词方法从理解句意的角度出发，达到识别出词汇的效果。这种方法的基本思想就是把句法和语法分析添加到分词的过程中，通过对文本的语义层理解，来消除歧义词这种现象。它的实现需要有一个统筹全局的总控，在

总控下，首先对文本进行分词，然后对分词结果进行句法分析，得到句子的语义信息，根据语义信息进行分词。分词过程中，文本深层语义信息的挖掘对分词的结果影响很大，由于中文在文本语义理解上有更高的复杂性，很难将文本语义以计算机公式的形式直接输出，因此目前还没有具备代表性的算法被广泛使用。

基于统计的分词方法是机器学习技术催生的一种分词法。它会学习已分词数据的规律，然后基于学习到的规律实现分词，这里学习规律的媒介就是机器学习模型。代表性的基于统计的分词方法有最大概率分词法 [19] 和最大熵分词法 [20]。用于学习分词规律的机器学习算法有 N 元文法模型 [21]、隐马尔可夫模型 [22]、条件随机场模型 [23] 和支持向量机 [24] 等。机器学习需要投入大量的训练数据，早期由于数据量少，模型训练效果一般，但随着可用语料库的不断扩充，基于统计的分词方法的分词效果也越来越好，现在逐渐成为了大多数研究人员分词的首选。在很多实用工具中，这种分词法同样要用到词典匹配，因为词典匹配的效率更高，此外还会结合统计学方法，以确保能够识别出生词并消除歧义词。现有的这类中文分词工具包括 THULAC(THU Lexical Analyzer for Chinese) 和结巴分词器 (jieba) 等。

本文使用的是 jieba 分词器，它不单独属于某一种分词方法，而是规则与统计方法的融合。jieba 会对词典中存在的词进行扫描，并构造前缀词典，形成层级包含结构，然后用动态规划算法，寻找概率最大的路径。如果某些词不在预设词典中，那么这些词的位置会被标记成一个训练集，用 HMM 模型和 Viterbi 算法训练它们，进而得到分词结果。jieba 有三种模式，分别是切分粒度最细致的精确模式、能识别出全部词汇的全模式以及适合查询的搜索引擎模式。本文选择能够准确切分词汇的精确模式把司法裁判文书中的词不重复地提取出来。文本关键词的提取也是 jieba 分词的一部分。关键词提取的实现基于 TF-IDF 算法，该算法计算分词后的每个词语在文本中的重要性。算法由词频和逆文档频率两部分组成，TF-IDF 值等于两者的乘积，其计算结果可以象征某个词在对应文本中的重要程度，即某个词在大多数文本中没有出现，但在某篇文本中有多处存在，那么这个词对于这篇文本一定是重要的。

$$TF_w = \frac{t_w}{\sum_{i=0}^{N} t_i} \tag{2.1}$$

$$IDF_w = log\frac{\sum_{i=0}^{M} d_i}{f_w\left(\sum_{i=0}^{M} d_i\right) + 1} \tag{2.2}$$

如公式 2.1， 2.2所示，TF 代表词语 w 在某篇文本中的频率，即该词语出现的次数 t_w 占所有词语数量的比例。IDF 代表包含词语 w 文本的比例，如果 TF 越大且 IDF 越小，则说明词语 w 的重要性越高。这就是 jieba 分词器关键词提取算法的基本原理。

2.1.2 词向量技术

词向量技术是中文文本分析最常用到的技术之一。之所以研究词语的向量表示，是因为词语是文本的最细粒度，可以最精确地描述文本特征。因此对于文本分析问题，究之根本就是要研究词语。

对于词向量 (word embedding) 的理解，本文基于有监督机器学习举一个判断词语词性的例子来说明。假设当前有一组 (x, y) 的词语标记样本，x 代表词语，y 代表词语对应的词性。我们想要构建一个函数，用于把词语映射到对应的词性上。但数学函数的输入通常都是数字，而中文文本不是数字，它是文字语言，如果想要计算机识别并处理文本，则要提前把文本通过技术手段变成数字，也就是把文字映射到数学空间中，这种映射的结果，就是词向量。本文用到的 Word2vec [25]，就是一种词向量方法。

下面将介绍应用范围广、且认可度较高的两种词向量表示法，分别是离散表示法和分布式表示法。

离散表示法的向量长度为词表大小，该词语出现在词表中的位置处的值置 1，其他位置均置 0,。例如，如果词表为 [西瓜，苹果]，词表长度为 2，则：

西瓜：[1,0]

苹果：[0,1]

但这种表示方法有一个明显的缺点，就是向量长度太长且太稀疏，导致特征向量空间会过于庞大，这也会导致词语之间的向量关联性变小，可能导致机器学习模型的能力变弱，很难从语义的角度保证模型质量。

分布式表示法则把词语定义成定长、连续且稠密的向量，它包括基于统计的生成方式和基于语言模型的生成方式两种分布式词向量生成方式。

本文使用的 Word2vec 就是一种基于语言模型的词向量生成方式。这种方式需要训练出一个语言模型，然后用语言模型对文本词语进行预测，训练过程中的中间隐藏层向量就是词向量。Word2vec 常用的向量有 50 维向量和 100 维向量，模型训练者也可以自定义向量维度。效果比较好的语言模型有 Skip-gram、CBOW、GloVe 和 FastText 等。

Word2vec 是有监督的机器学习模型 [26]，它符合 $f(x) - y$ 这一映射。x 是文本中的某个词语，y 是 x 的关联词语，f 就是语言模型，这个模型的目的就是判

断 (x, y) 这个词对，是否符合自然语言的法则。Word2vec 并不是为了把语言模型训练得多完美，而是为了得到模型参数，进而用向量的形式来刻画 x。

Word2vec 有两种语言模型，分别是 Skip-gram 和 CBOW。Skip-gram 用一个词作为输入，来预测这个词的上下文，例如对于一句话 “张某被处以 50000 元罚款”，输入是 “张某”，输出可以是 “元”、“罚款”；而 CBOW 则相反，它的输入是某个词语的上下文，而输出则是一个被预测的词。

在 Skip-gram 模型中，词语 x 的原始输入是一个唯一的向量，这个向量在 x 出现的位置置 1，其余位置全置 0。例如 “张某 被 处以 50000 元 罚款” 这句话，共有 6 个单词，那么向量就是 V=6 维的，“张某” 是第一个单词，则把第一个位置设为 1，其余位置为 0。

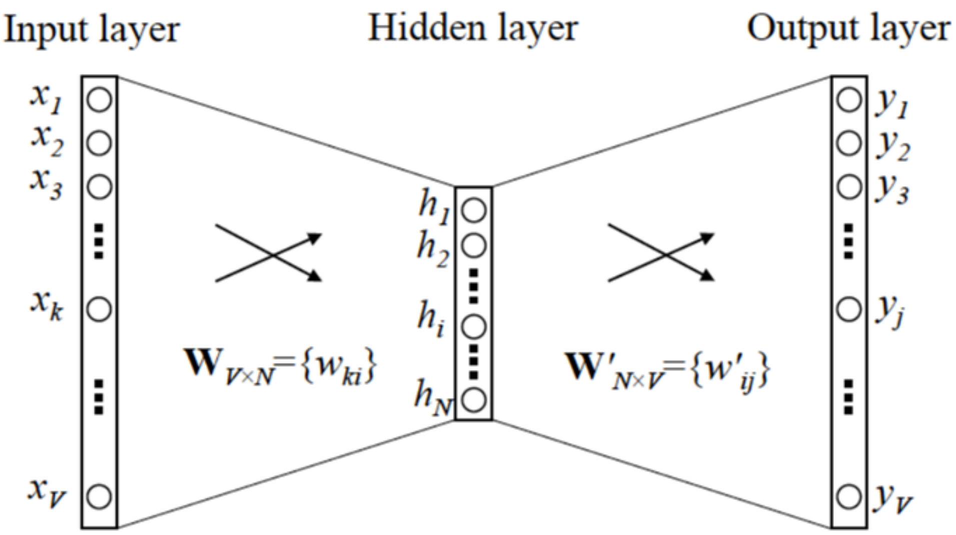

图 2.1: Skip-gram 的简单网络结构

图 2.1为 Skip-gram 的简单网络结构，x 就是前面所说的原始输入，y 是每个词语的输出概率。这里隐藏层的激活函数是线性的，用反向传播算法（即链式求导）来训练神经网络。训练后得到神经网络的权重，例如输入的 x 为 [1,0,0,...,0]，那么从输入层到隐藏层的权重里，只有第一个位置的权重被激活，权重的个数和隐藏层的节点数是一致的，进而由权重组成一个向量 v_x 来表示 x。因为每个词的原始输入是唯一的，所以这个词对应的向量 v_x 也是唯一的，因此这个向量就可以用来唯一表示这个词语。

图 2.1展示的结构是最简单的 Skip-gram 结构，即输出 y 只有一个单词，当输出有多个单词时，Skip-gram 结构如图 2.2所示。此时的结构可以看作 x 到 y 的若干个并联。

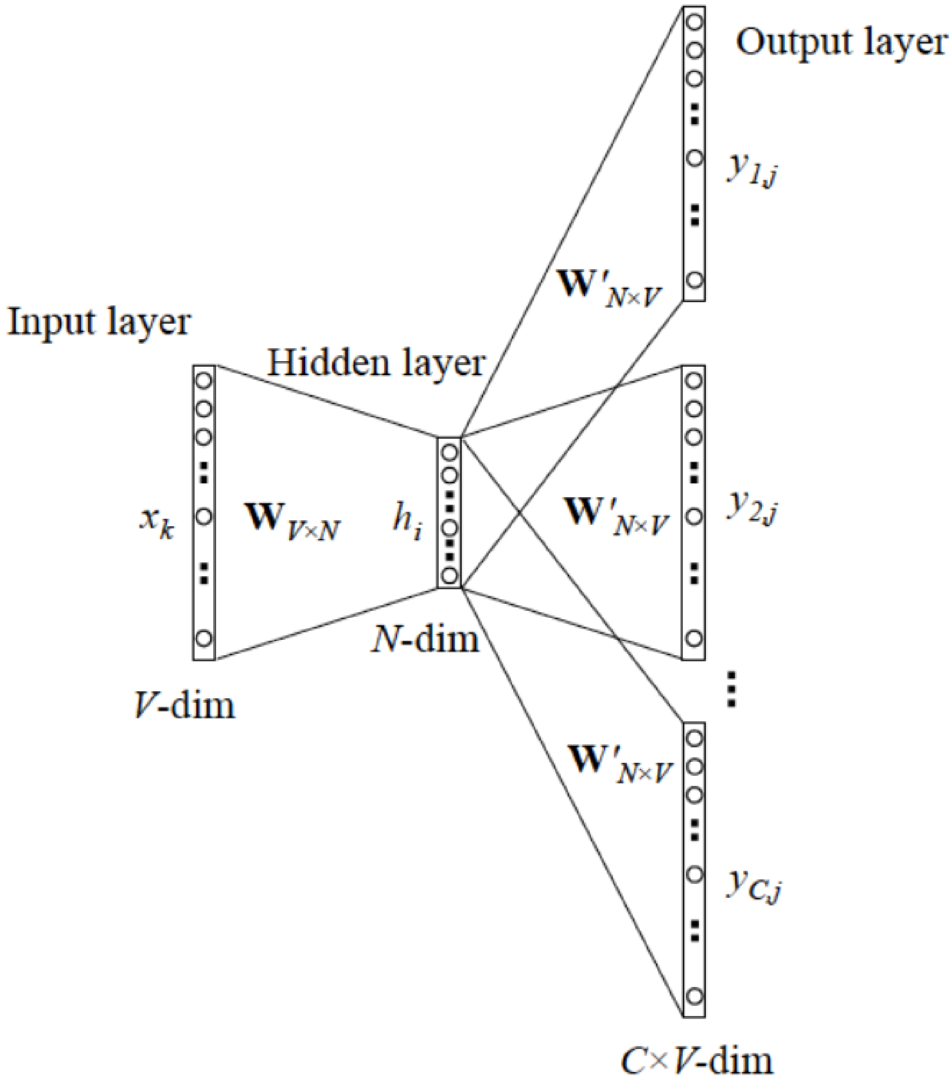

图 2.2: Skip-gram 网络结构

CBOW 和 Skip-gram 类似，只不过 CBOW 是通过上下文来预测一个词语。如图 2.3所示,CBOW 的输入为多个词语，一般要对多个输入求和后取平均。

在词向量技术中，词向量的维度与隐藏层的节点数一致，这个数字通常比总词汇量 V 要小很多。因此 Word2vec 实质是一个降维操作，它将很高维的词向量转化成较低维的词向量，降维不仅可以减少存储和计算空间的损耗，也可以提高模型训练的效率和质量。

在近些年关于词向量技术的应用中，已有研究表明，Word2vec 的 CBOW 模型在司法领域的机器学习预测算法上有优异的表现 [27]，因此，本文也采取 CBOW 作为语言模型。

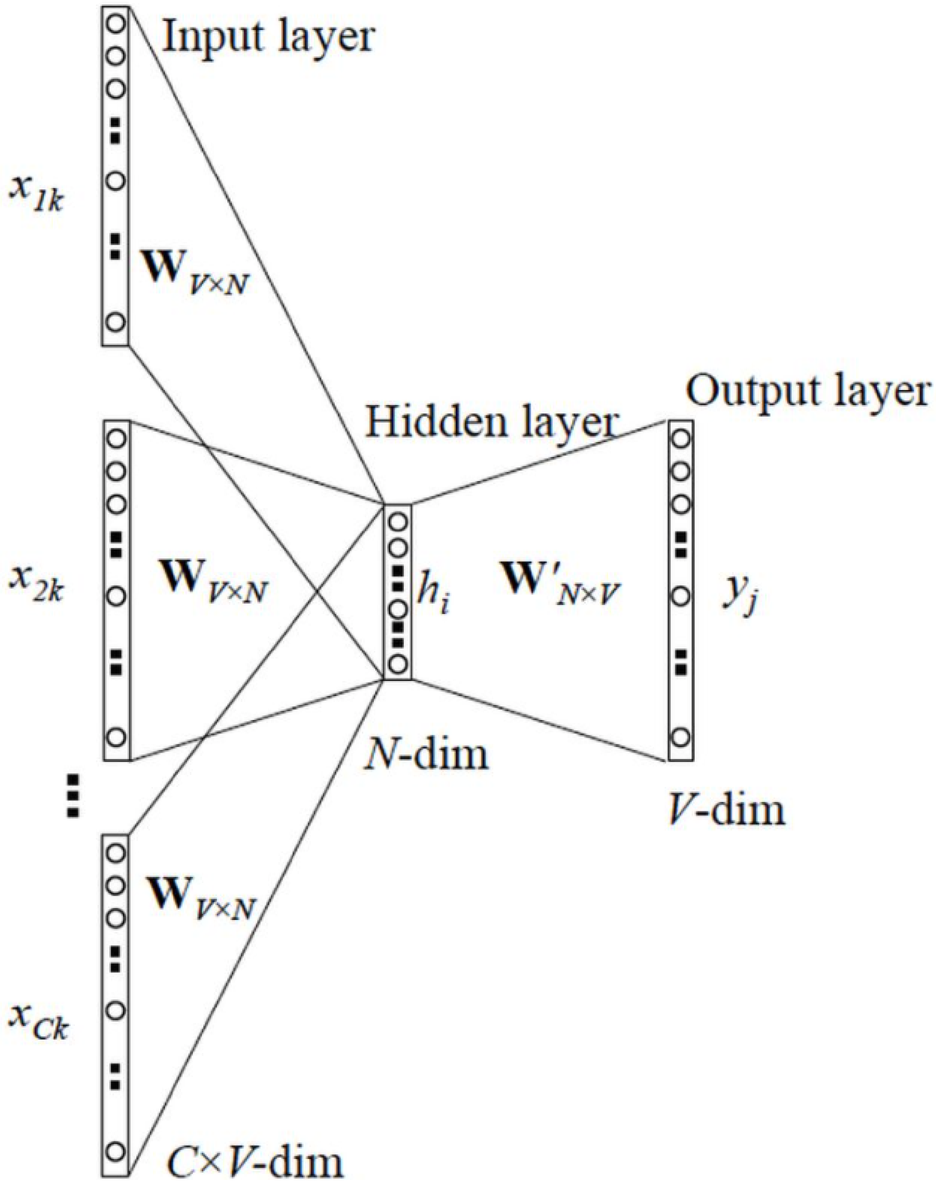

图 2.3: CBOW 网络结构

2.1.3 文本情感分析技术

司法文本是司法案件的客观过程描述，高质量的文本应当具备情感中立性，不应该带有法官及司法裁判文书记录员的主观情感。如果要扩增出高质量的司法文本，文本情感分析技术也是必不可少的一部分。

文本情感分析 (Sentiment Analysis) 技术是一种较为复杂的综合性自然语言处理技术，通常要结合文本挖掘共同实现，它的主要目的是研究文本的主观情感特征。对其研究对象的粒度进行分类，可以划分为词级、句级和篇级；根据研

究的行业不同，可以划分为各个领域的特征研究；根据任务的不同，可分划分为情感分类、情感检索和情感抽取等类别。文本情感分析流程如图 2.4所示，整个过程包括了文本预处理、构建词库、情感分类、情感检索和情感抽取等步骤。

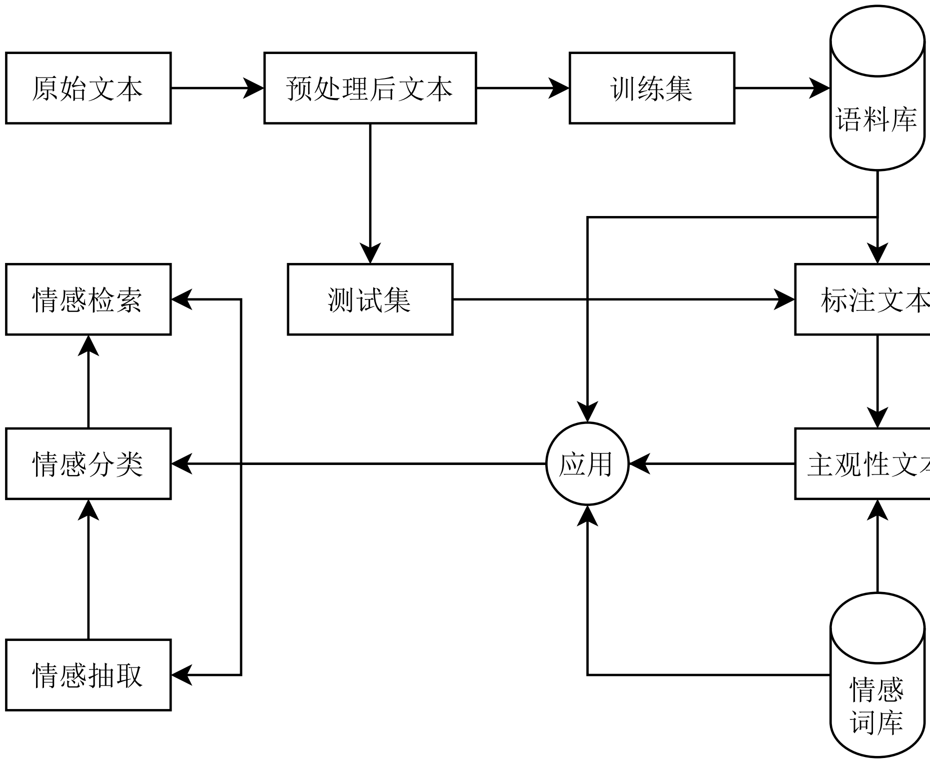

图 2.4: 文本情感分析流程

情感分类是研究情感倾向性的技术，对于一段文字，需要识别出文字的情感倾向性是肯定还是否定的，即是正面还是负面的，这也是情感分析技术最主要的研究方向。从广义上看，所有文本类数据都属于主观性文本或客观性文本。主观性文本带有文本创作者的个人主观情感，而客观性文本则是文本创作者对客观性事实的描述。情感词是判别文本主客观偏向的主要依据，分类算法则用于实现最终的情感判断。

情感分类有两种主要的技术路线，一是利用情感词典进行分类，二是利用机器学习技术进行分类。

利用情感词典进行分类的方法，首先需要构建一个词语与情感评分对应的词典，在词典中，每个词都有一个分数用于描述它的情感偏向以及偏向程度。有了词典后，还需要一个基于统计学的算法，用来计算文本的得分。最后根据得分判定文本的情感偏向。

根据上述步骤可以得知，构建情感词典是分类过程中至关重要的一步。因

为国内外对情感词典的研究已经较为成熟，所以新涌现出来的词典也基本都是对现有词典的扩充。在英文词典方面，WordNet 的认可度最高，很多研究人员都会在它的基础上构建自己的情感词典 [28]。中文词典方面，Hownet 是很多专家学者的首选，对于不同的研究领域，可以构建具备行业特色的情感词典，虽然这样工作量会增加，但是对于提高分类的准确率有很大的帮助。

基于情感词典构建一个统计学算法也很关键，这是计算文本情感得分的核心步骤。该算法应当采用加权求和的思想，对词典中不同类的词区别对待，设计合理的权值分配方案以及权值计算方法。这里介绍一个通用的权值计算方法，如公式 2.3所示，N_p，N_n 分别表示正面和负面情感词语的总词汇量，wp_i，wp_j 分别表示正面和负面情感词语的权值。采用这种加权平均算法，一定程度上可以提高情感分类的准确率。

$$E_{avr} = \frac{\sum_{i=1}^{N_p} wp_i + \sum_{j=1}^{N_n} wp_j}{N_p + N_n} \tag{2.3}$$

构建完成统计学算法后，就需要根据文本得分来预测情感偏向了。通常我们规定，如果加权算法的计算结果为正数，则说明文本是正向的；如果加权算法的计算结果为负数，则说明文本是负面的；如果加权算法的计算结果为 0，则说明文本是中立的。对于算法效果的评价，一般采用统计学中的正确率、召回率和 F 值作为指标。这种利用情感词典进行分类的方法，属于一种粒度较粗的情感分类方法，但它不需要标注好的大数据集，所以可操作性更强，成本也更低，可以快速对文本数据集进行情感分类。

另一种情感分类方法是利用机器学习技术进行分类。主题模型是一种常见的分类模型，它会学习并提取文本的主题，以主题为分类标准，实现文本的归类，不过这种方法是黑盒性质的，主题也是无法人工指定的，具备一定的不确定性。在应用主题模型时，通常会指定某个领域，把文本聚焦在该领域上，例如新闻、体育、金融。文本情感分类与主题模型分类有相似的地方，但也有不一样的地方。文本情感分类不在乎文本内容是什么，它根据文本本身的情感特性完成情感判断。

包括主题模型在内，很多机器学习分类方法都可以直接、或者通过改进成新方法间接应用到情感分类问题上。其整体流程为：预先标注文本，然后将标有情感标签的文本投入训练模型，获得能够识别文本情感偏向的机器学习模型，最后把测试文本数据放到情感分类器中，得到文本的情感倾向结果。训练得到的机器学习模型之所以能够对文本进行分类，是因为模型掌握了文本的情感特

征。常见的获取情感特征的方法有信息增益法，CHI 统计量法和文档频率法等。获取到文本情感特征后，就可以用中心向量法、K Nearest Neighbor(KNN)、贝叶斯分类法等方法训练得到机器学习模型了。Pang 等人 [29] 提出了一种准确率较高的情感分类方法，这种方法借用了词袋的思想，即所有词语无差别地装在一个袋子中，用机器学习模型去学习已标记文本的情感特征规律，实验中使用的算法是支持向量机，研究证明这种方法的分类准确率能够超过 80%。通常情况下，对于文本的情感分类，还达不到人工情感判断的精确度，这是由于语言学中复杂的文字表达以及文字的多层含义还不能让机器完全学习。在基于机器学习的情感分类算法中，情感特征的选择对最终情感分类的效果影响很大，很多研究者都把情感分类看成一个特征优化问题去研究 [30]。越来越多的研究者参与到情感分类的科研中，已有的情感特征也正在被逐步扩充，语料库也随之越来越庞大，基于机器学习的情感分类技术也会有更广阔的发展前景。

文本情感分析技术中，研究最多且应用最广的是情感分类，此外，情感检索和情感抽取也是较为重要的研究方向。情感检索是比情感分类更为复杂的任务，它需要先获取指定的信息，然后对获取到的信息进行排序，排序的依据是主题相关性和情感偏向性。情感检索可以通过检索算法筛选出主题相关的文本，然后对筛选的文本进行情感分类，也可以自行设计算法，把主题相关性和情感偏向性量化，通过计算筛选出符合要求的文本。情感抽取的任务是在带有情感偏向的文本中，获取蕴含主观情感的词语，或者识别表达情感的主体。情感抽取是一些情感分析任务的基础，如果情感抽取工作做得好，那么对下游很多应用的高效实现会有推动作用。

2.2 数据扩增技术

本文所提出的数据扩增技术聚焦在司法领域，既要用到传统数据扩增的方法，也要结合司法行业的特殊性，这样才能够保证扩增出高质量的司法文本。

2.2.1 通用数据扩增技术

数据扩增 (Data Augmentation) 是一种基于已有数据，通过各种形式的数据处理，生成新数据的技术。它解决的问题是：在机器学习过程中，缺少足够的数据，导致训练出来的模型过拟合化。根据扩增对象的不同，数据扩增技术可以分为图像扩增和文本扩增。顾名思义，图像扩增的对象是图像，而文本扩增的对象是纯文本。

图像扩增的主流方法有七种，分别是翻转 (Flip)、旋转 (Rotation)、缩放比例 (Scale)、裁剪 (Crop)、移位 (Translation)、高斯噪声 (Gaussian Noise) 和条件对抗

神经网络 (Conditional GANs)。

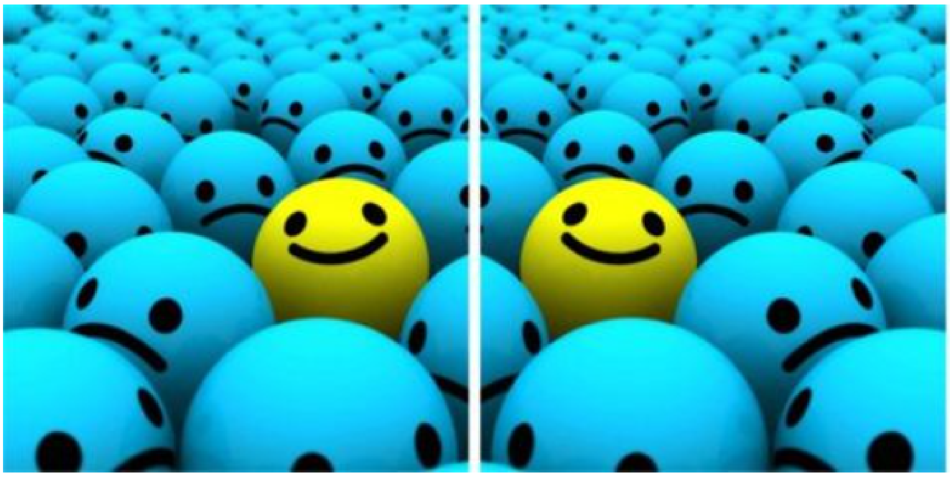

图 2.5: 翻转扩增

图 2.5为翻转扩增的示例，翻转有两种，水平翻转和垂直翻转，图 2.5左侧为原图，右侧为水平翻转。

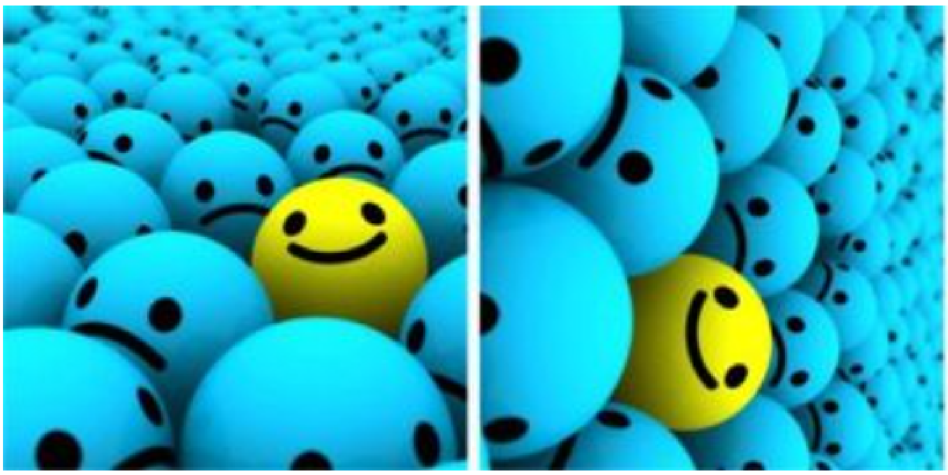

图 2.6: 旋转扩增

图 2.6为旋转扩增的示例，旋转扩增分为顺时针旋转和逆时针旋转，图 2.6左侧为原图，右侧为顺时针旋转 90 度。

图 2.7为缩放比例扩增的示例，缩放比例扩增分为向内缩放和向外缩放，向外缩放将会放大图像尺寸，向内缩放则会缩小图像尺寸。

图 2.8为裁剪扩增的示例，裁剪扩增和缩放不同，裁剪扩增是从原图中随机取一个部分，然后将此部分调整为原图大小，对比图 2.8和图 2.7可以看出区别。

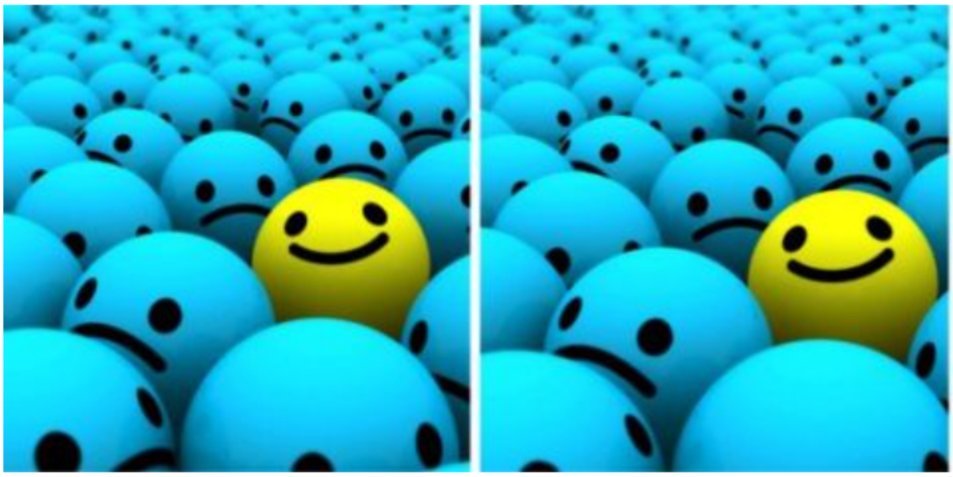

图 2.7: 缩放比例扩增

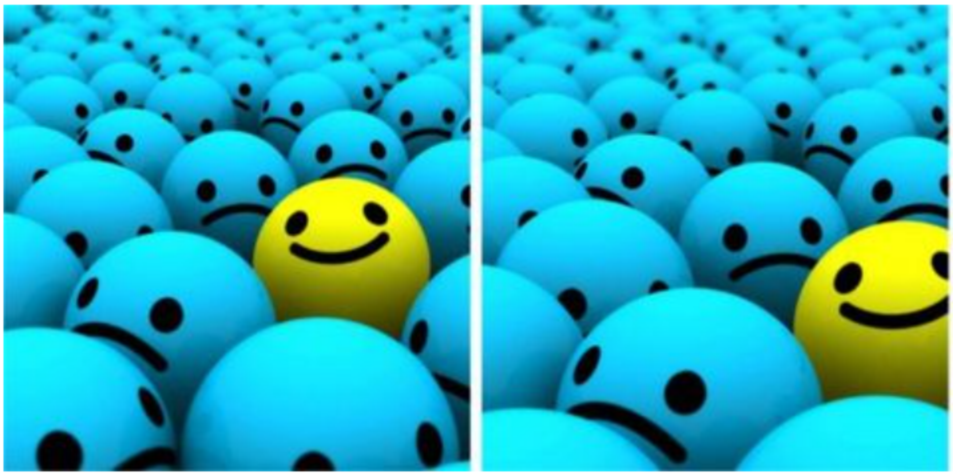

图 2.8: 裁剪扩增

图 2.9: 移位扩增

图 2.9为移位扩增的示例，移位扩增可以分为水平移位和垂直移位，图 2.9左侧为原图，右侧为水平向右移动。

基于高斯噪声的扩增就是为原图添加一些高斯噪声类型的像素。如果神经网络学习的数据含有无用的高频特征，那么就会造成模型过拟合的后果。添加 0 均值高斯噪声能够减小过拟合的程度，但如果噪声添加过量，则会影响到数据质量，不过通过某些技术手段也能够减轻这一影响。因此我们可以得知，高斯噪声的掺入要适量，否则会影响模型的扩增能力。

基于条件对抗神经网络的扩增是一种更高级的扩增技术，它可以改变图像的风格。在现实中，有很多室外照片和风景照片，它们可能是峭壁、海洋或者森林，但是很多情况下，我们可能只有景观在某一时刻的图像数据，这时候如何扩增出一年四季的景观图像就成为了难题。如果扩增模型不知道同一景观在不同天气、季节状况下的特征，那么峭壁也可能被识别为山峰。面对这类问题，最简单的解决办法就是继续添加包含各种天气、季节条件下的景观图片，但这个方法的工作量太大，极为繁琐。这时候就可以用到条件对抗神经网络，条件对抗神经网络可以将图像从一个域转换到另一个域 [31]，例如可以将夏季风景照转换为冬季风景照。

文本扩增是与图像扩增相并列的一种扩增技术，虽然只是扩增对象发生了变化，但是两者的扩增方法却相差很大。通用的文本扩增方法有：回译、同义词替换和基于深度学习的扩增方法。

回译就是先把中文翻译成外文，然后再翻译回来，常用的回译外文有英、法、德、俄、西班牙、葡萄牙、日、韩、荷兰、阿拉伯等语言，这种方法适合扩增单篇文本，扩增的单篇文本自身数据质量较高，但是很难扩增出数量更多的文本数据。当前可调用的回译接口包括谷歌翻译和百度翻译等等。

同义词替换就是随机或主动选择文本中的某些词语，在已有的词库或自主构建的词库中选择合适的同义词来替换，从而扩增出新的文本数据。EDA[32] 是现有的较为成熟的同义词替换方法，它通过替换、插入、交换和删除等操作来实现数据扩增。

基于深度学习的扩增方法就是训练出能模拟人思维模式的扩增模型，利用这个模型来实现文本的扩增。例如根据文本的标签信息以及 bi-LSTM 实现深度学习模型的训练，然后用训练后的模型完成数据的扩增，又如用 Word2vec 词向量技术来训练深度学习模型，以用于数据扩增。

2.2.2 司法文本数据扩增技术

司法文本数据扩增技术和通用数据扩增技术有相似的地方，但也有一定的区别，区别在于司法行业的特殊性。司法裁判文书有特定的结构规范和质量评价体系，在司法领域应用的数据扩增技术必须带有明显的行业属性。

Guo 等人提出的 TauJud[7] 是近几年最好的司法文本数据扩增技术之一。TauJud 可以快速扩增出大量测试数据，并且提供了司法领域的数据扩增工具。

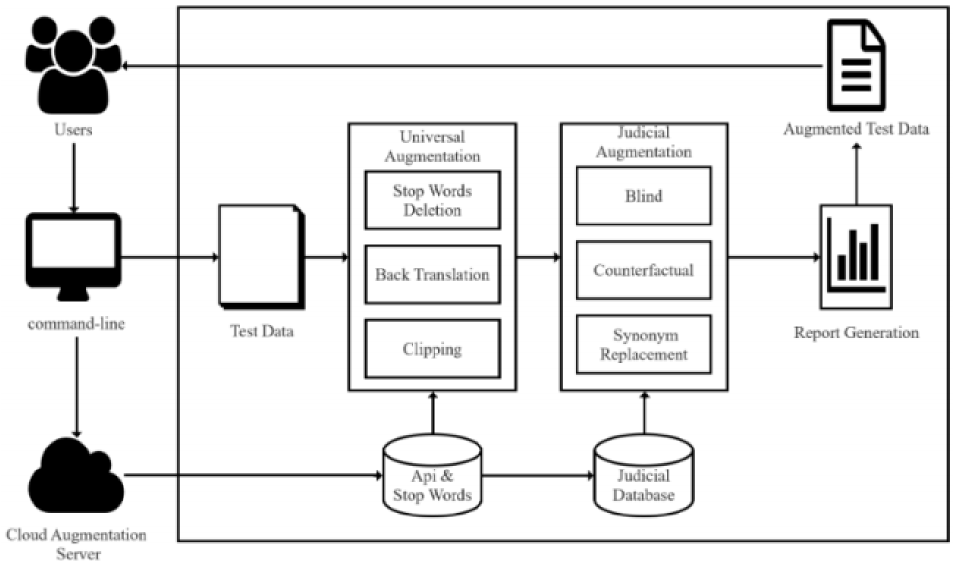

图 2.10: TauJud 工作流程

图 2.10是 TauJud 的工作流程。作为一个数据扩增工具，它的输入为原始司法数据，当司法数据传入流程时，服务器上部署的扩增任务立即被启动。首先执行的是通用扩增，具体包括了停用词删除、回译和文档裁剪。通用扩增之后是司法扩增，这是针对司法数据具备的领域特征所执行的扩增。完成上述扩增任务后，TauJud 还可以提供一份扩增结果报告。TauJud 是基于通用扩增方法的改进，并没有考虑太多扩增后数据的质量好坏，这可能导致扩增文本的质量良莠不齐。因此从质量维度出发，进行司法文本数据的扩增，可以达到更好的扩增效果。

2.3　数据质量评估技术

数据质量的定义最早由 Richard Y 等人 [33] 提出，他们认为数据质量是数据满足使用需求的程度。此外，Richard Y 还指出数据质量的评价应当由数据使用者做出，他们列举出了一组属性，这些属性可以代表数据质量的好坏，这就是数据质量评估理论最早的框架基础。

数据质量评估指标的提出和完善经历了一个较为漫长的过程，Aebi 等人 [34] 提出了准确性、一致性和完整性等评估指标用于描述数据质量，但是研究只停留在定性层面，没有完成定量评估的工作。在通用质量评估技术中，Wand 等人 [35] 提出了基于本体论的数据质量评估体系，Wang 等人 [36] 借鉴数据工程的相

关指标，将数据质量划分为主观和客观两个维度。在二十世纪末，Tate 等人 [37] 针对网络数据提出了准确性、权威性、客观性等六个指标。Zhu 等人 [38] 于 2000 年也提出了六个指标，与之前不一样，这些指标是针对信息检索系统的，包括流通性，信息噪声比、流行程度等。分析这些研究成果，可以发现不同学者提出的数据质量评估体系存在一定的差异，针对这个问题，Knight 等人 [39] 归纳了常见的质量维度，并统计了不同维度在各个评估框架中的出现频率，经过统计，最常用的数据质量评估指标如表 2.1所示。

表 2.1: 数据质量评估指标统计表

指标	**出现次数**	**指标含义**
准确性	8	数据正确、可靠并且无误
一致性	7	数据以相同格式呈现并与以前数据兼容
安全性	7	为维护信息的安全对信息的访问控制设置
及时性	7	对于当前任务，信息是否足够新
完整性	5	数据结构完整，并且能够满足使用者对数据的需求
简洁性	5	数据在表述的过程中言简意赅

我国关于数据质量的研究，起步要远晚于国外。直到 2008 年，才由部队研究所提出了一些数据质量相关的研究问题，包括如何定义数据质量，如何定义错误源，如何改进数据质量等 [40]。中国科学院于 2009 年提出了一套较为完整的数据质量评价体系，这套体系不仅包括了数据质量的度量方法，还讲解了数据质量度量的相关指标含义。Cai 等人 [41] 从使用数据的用户的视角出发，构建了包括大数据质量、质量指标等在内的多层次数据质量评估体系，并提出了动态评估的方法。

信息质量在一定程度上也可以归类为数据质量。信息是数据的抽象，可以理解为数据所表达的语义内涵。之所以研究信息质量，是因为质量差的信息可能导致严重的事故，例如商业订单的错发、电信票据的误识别、导弹的脱轨等。因此信息质量的研究在很多领域都非常重要。

信息质量的评估有两种主流方法，分别是基于数据库的方法和基于管理的方法。基于数据库的方法需要足够数量的数据，根据数据的特征进行信息质量的评估。数据库派是指在基于管理的方法则是基于工作业务，通过分析数据的流转以及交互来实现信息质量的评估。因为本文研究的对象是司法裁判文书，属于现有数据，所以本文对信息质量的评估会使用基于数据库的方法。

图 2.11是信息质量评估框架示意图，这个框架由 Su 等人 [42] 提出。可以看到，框架总共分为三个部分，也可以看作三层，从下到上分别是度量层、维度

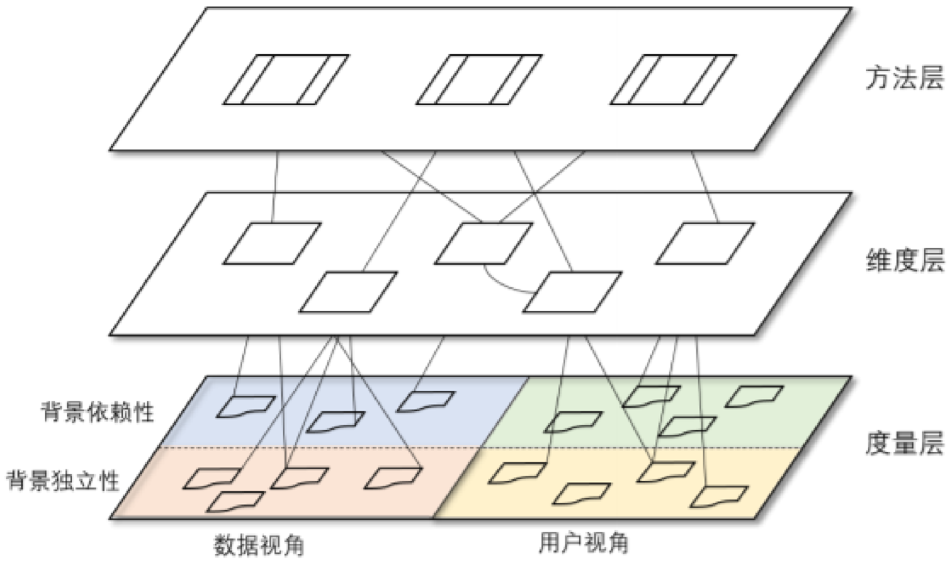

图 2.11: 信息质量评估框架

层、方法层。最下层为度量层，它代表了数据视角和用户视角，该层度量数据的方法需要基于背景独立性和背景依赖性；第二层为维度层，这一层有很多信息质量相关的维度，每个维度都有量化的指标，这里要注意的是，维度与指标之间是多对多的；最上层就是方法层了，方法层和评估信息质量的各种模型相关，更偏向于结构化的部分。该信息质量评估框架可以应用到各种各样的信息质量评估场景中，如果想得到更精确的评估结果，则需要根据场景的不同，定制化信息度量的维度。本文的研究对象为司法文本，在司法领域，数据质量和信息质量都是极其重要的部分，本文结合两者的特点，把信息质量归到数据质量中，进而扩增出高质量的司法文本。

2.4　web 应用框架 Django

本文提出的司法文本数据扩增技术，需要依附于一个数据扩增系统来实现。如果要设计一个系统，就必须考虑到系统的可用性和交互友好性。当下很流行的 B/S 架构就具备了开发出可用且易用系统的特质。B/S 是指浏览器和服务器，在 B/S 系统架构中，访问者通过浏览器申请访问系统。web 应用框架 Django[43] 就是一种适合搭建 B/S 架构的技术框架。Django 的开发语言是 Python，它是一个开源的网络框架。它最大的特点就是易拓展、易管理和易维护，同时还具备了解决网络交互问题的能力。使用这个框架，开发者不用过多关心底层代码的实现，只需要专注于上层 web 应用的构建。

在 web 应用的开发中，MVC 是一种较为流行的设计模式。其中，M 指的是

Model，它负责数据和业务的处理；V 指的是 View，它负责与用户交互；C 指的是 Controller，它负责将用户请求和业务逻辑相关联。这种模式把 Model, View, Controller 以松耦合的方式连接，保证了三者相对独立。而 Django 不属于 MVC 模式，它属于 MTV 模式，其设计思想虽然和 MVC 类似，但是整体模式和功能还是有所区别的。MTV 模式功能如表 2.2所示。

表 2.2: MTV 模式功能

层级	**功能**
模型 (Model)	负责与数据库的连接与交互，包括数据库表的定义、数据的存取、数据的验证等操作
模板 (Template)	负责与页面的展示及与用户的交互，定义页面的展示模板，及业务逻辑写入页面模板等操作
视图 (View)	负责业务逻辑相关操作，调用 Model 和 Template 的内容

接下来详细介绍 Django 框架的工作流程：

用户申请访问某个服务页面的 url，服务器收到请求后，把这个请求发送到 WSGI(Web Service Gateway Interface)，它负责定义网络应用与服务器之间的接口，以确保网络应用和服务器之间可以正常通信。服务器会接收到来自用户的 http 报文请求，然后将该报文转换成 WSGI 格式，之后再发送给应用服务。将 http 报文转换成 WSGI 格式的处理函数会响应两个参数，第一个参数是字典格式的，字典中涵盖了请求信息（例如请求路径、请求方式），第二个参数是调用的函数，这个函数包含 headers 和状态码等信息，它用于发起响应。

用户访问某个 url 时，需要 URLConf 来为用户寻找 View，用户的访问位置会在 view.py 中被定位，然后调用对应的函数。urls.py 负责存储 Django 框架下的 url 映射器。url 映射器中定义了一个映射表，这个列表中存储了各个视图列表以及它们对应的路由，如果收到了某个请求，则按照规定的顺序，在列表中一个个匹配，一旦匹配成功，就开始调用相应的视图函数，此时该函数会收到 request。

视图负责处理请求函数。它是 web 架构中最核心的部分。在接受到请求后，它会返回响应。视图通过模型访问数据库，获取所需要的数据，之后通过逻辑函数的处理，得到响应对象，并将它传递给模板。模型负责管理应用程序，以及数据库相关的操作。模型可以定义数据结构，例如字段类型、字段名、默认值等。模型对数据库的定义和基础数据库有所区别，Django 有专门的数据库查询 API，因此在视图函数中，不用直接和数据库进行交互。

响应对象会被传输到相应的模板中，用来确定模板中的变量，进而渲染出交互页面。Django 的模板渲染流程如下：先创建一个模板对象，然后用视图中

的逻辑函数来传递数据，这里要用到 context 类，context 由字典映射变量和填入变量的值共同构造而成，模板调用方法 render，把 context 写入 html 文件。

2.5 分布式存储

伴随着信息化时代的到来，信息数据迎来了爆发式的增长，传统的数据库也面临着越来越大的挑战，数据存储也随之有了更高的性能要求。传统的数据存储方式是集中式，这种架构存在空间有限、拓展不灵活、无多点容错等问题。分布式存储技术的出现极大缓解了这些数据存储的需求痛点。分布式存储的设备从单一的计算机变为了分布式的计算机集群，将数据存储到分布式节点上。这种方式有效地缓解了存储空间小、拓展性差的问题，也提高了存储系统的容错率。对于分布式存储系统而言，如何解决数据的可用性、一致性和分区容错性的问题，是最为关键的部分。

Hadoop[44] 是著名的分布式存储架构，它由 Apache 开发。Hadoop 可以将海量数据存储到分布式节点中，并可以直接在节点中完成数据计算任务。它仅使用了很简单的模型，就可以对数据进行跨机的处理。面对存储服务器可能遇到的突发故障，Hadoop 设计了完善的故障检测和处理机制，使得整个系统可以持续健康工作。

Hadoop 分布式文件系统 (HDFS) 是 Hadoop 的底层存储系统。HDFS 属于分布式存储结构，它将文件存储于各地不同的服务器节点中。它带有心跳检测机制、副本存放策略和流式访问模式，这些机制使得 HDFS 具备很多特有的优势，例如高可靠性、高容错率和高速计算的能力。

HDFS 可以分为四个模块，分别是 NameNode、Secondary NameNode、DataNode 和 HDFS client。各模块关系如图 2.12所示。

在 HDFS 中，HDFS client 是用来访问 HDFS 服务的客户端，它通过预定义的指令与 HDFS 产生交互，这里的交互包括启动、访问、管理和关闭等。

NameNode 负责和客户端的读写请求进行交互，它提供文件的位置信息给客户端，并负责管理副本策略。NameNode 和数据节点的连接方式是松耦合的，HDFS 可以采用动态适应的调节方式来增加或减少集群节点。

DataNode 需要保持与 NameNode 的通信，以查询是否有待执行的任务。持续通信是为了保证每个 DataNode 都在持续地被监控。DataNode 会执行接收到的读写命令，并把文件分布式地存储到不同的块，也可以从不同的块读取文件。如果某个 DataNode 出现了故障，无法继续工作，那么 NameNode 会立刻将任务重新分配给其他包含相同数据块的节点。

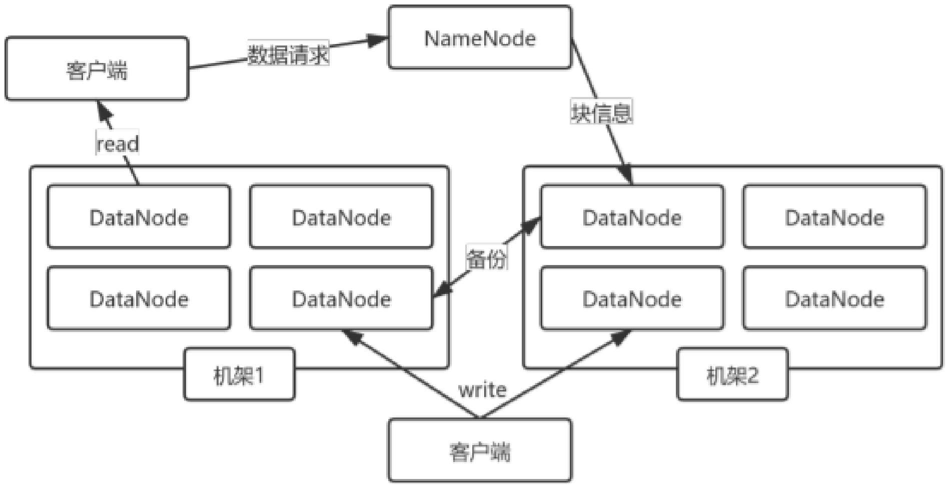

图 2.12: HDFS 模块关系图

Secondary NameNode 可以看作是 NameNode 的辅助，它会协助合并文件信息，并将合并后的文件信息传送给 NameNode。

2.6　本章小结

本章介绍了本文用到的相关技术，包括自然语言处理技术、数据扩增技术、数据质量评估技术、web 应用框架 Django 和分布式存储技术。其中，自然语言处理技术部分又分别介绍了中文分词技术、词向量技术和文本情感分析技术，这些自然语言处理技术为本文的文本处理过程提供了理论依据；数据扩增技术分为了通用数据扩增技术和司法文本数据扩增技术，从通用领域和司法特征领域两个角度引出了本文所用到的数据扩增技术；数据质量评估技术部分介绍了本文用到的数据质量维度；Django 这一部分则简述了本文系统实现所使用的网络架构；最后的分布式存储部分说明了本文的数据存储方式和计算方式。

第三章 需求分析与概要设计

本章将会分析系统的需求，并完成系统概要的设计。首先会从功能和非功能的角度分析本文设计的数据扩增系统有哪些需求，然后基于需求分析，给出系统的概要设计，并将系统划分成基本的功能模块，之后对四个主要的功能模块拆分介绍。

3.1 系统整体概述

当前机器学习技术正在快速发展，在智慧司法的背景下，机器学习也被用到了司法推荐系统中。但是因为司法文本的数据数量不够，且数据质量良莠不齐，导致训练出来的推荐模型质量不高。针对这种问题，本文设计了数据质量驱动的司法文本数据扩增系统。该系统基于常见的司法文本数据质量维度，结合通用领域的数据扩增方法，将词向量等机器学习技术加入到扩增过程中，能够为用户扩增出高质量的司法裁判文书。

本系统中数据质量驱动的司法文本数据扩增流程如下：总体分为客观质量维度的扩增和主观质量维度的扩增。首先解析用户上传的司法文本，包括结构化的标签提取和中文分词；将解析后的数据传入训练好的 CBOW 词向量模型，该模型是基于 Word2vec 方法训练出来的司法领域词向量模型，它可以根据词向量相似度计算方法，将输入文本中的词语替换成其他相似词；但是直接替换可能存在替换词带有主观情感的情况，这在司法文本中是不允许的。因此还要基于已有的司法文本训练出一个能体现词语中立性的情感词典，将这个词典和已有的中文情感词典 Hownet 合并，合并后的词典会标记很多带有主观情感的词语，在 CBOW 词向量模型中将这些词语设为禁用词，这样就可以保证 CBOW 模型的替换词都是中立性质的。以上部分为主观质量维度的扩增；接下来会进行客观质量维度的扩增，这部分根据文本预处理中提取的结构化标签，结合司法文本数据质量维度，进行文本的真实性扩增、一致性扩增和延迟性扩增等一系列质量维度驱动的扩增。以上两部分共同组成了数据质量驱动的司法文本扩增全过程。

根据以上思路，基于本文技术所开发的系统需要满足以下基本需求：管理用户的需求（包括用户注册、登录和管理等）、数据交互的需求、司法文本处理的需求、数据扩增的需求和文件数据管理的需求。根据这些需求，将整个数据扩增系统划分为四个模块，分别是数据交互、文书解析、数据扩增和文件管理。

各个模块之间的调用关系为：用户登录进入系统主页面，上传待扩增的司法裁判文书，此时系统调用数据交互模块，上传司法裁判文书。然后系统调用文书解析模块，对文本进行分词并提取标签。接下来数据进入数据扩增模块，这是系统中最核心的部分，在这个阶段，系统基于主观和客观质量两个维度，完成数据的扩增。最后，再次调用数据交互模块，下载扩增后的文本文件至本地。

3.2　系统需求分析

3.2.1　功能性需求

系统的功能性需求主要从数据交互、文书解析、数据扩增和文件管理四个模块进行分析，具体的功能性需求如表 3.1所示。

表 3.1: 功能需求列表

需求 ID	**需求名称**	**需求描述**
R1	上传司法数据文件	用户在进行司法文本的数据扩增前，预先上传司法裁判文书文件。
R2	下载司法数据文件	系统完成数据扩增后，用户选择并下载相应的文本文件。
R3	文书解析	用户上传文件后，系统自动将文件内容转换成 json 格式，并进行中文分词，然后提取文书相关标签并保存。
R4	主观质量维度的扩增	系统对解析后的文书进行主观维度的扩增，包括司法语言特征扩增和基于文本情感分析的扩增。
R5	客观质量维度的扩增	系统对主观维度扩增后的文本进行客观维度的扩增，包括延迟性扩增、真实性扩增和完整性扩增等。
R6	数据扩增文件的管理	用户在此页面查看和管理系统历史扩增的文件。

数据交互模块的主要功能是文件的上传和下载，具体来说就是用户上传待扩增的文件，扩增完成后，系统展示扩增后的数据文件，用户可以选择下载。该模块需要建立起和 HDFS 的数据交互接口，以便文件数据存储在分布式的集群上。

文书解析模块的功能是数据格式的转换、中文分词和标签提取。系统会把用户上传的数据，解析成系统统一的格式，然后用 jieba 进行分词，完成数据的预处理，最后进行数据标签的提取，得到数据标签对。

数据扩增模块是系统的核心部分，数据扩增功能就在这里实现。整个扩增过程分为主观维度和客观维度的扩增，通过语言特征扩增、情感分析、延迟性扩增等具体方法完成扩增任务。

文件管理模块的功能是对系统扩增出司法文本文件进行管理，具体包括文件的展示、文件的下载和文件的删除。

系统的各个功能模块之间是低耦合的，因此可以较为方便地进行修改和迭代，各模块之间的互相调用也很简便。

3.2.2　非功能性需求

非功能性需求是对功能性要求以外部分的补充。它虽然不是系统工作流程的描述，但同样不可或缺。非功能性需求用于约束系统的健壮性与鲁棒性，它对软件系统的性能提出要求。本系统的非功能性需求包括可靠性、安全性和容错性三个方面。具体如表 3.2所示。

表 3.2: 非功能性需求列表

非功能性需求	具体内容
可靠性	用户误操作时系统不崩溃，且返回操作提示。
安全性	数据扩增文件的查看、操作权限只限于文件作者和系统管理员。
容错性	用户数据不会因为服务器损坏而丢失。

可靠性表示系统对用户操作的响应应当是可靠的。如果用户没有按照系统规定的方式进行操作，那么系统不应该因此宕机或停止运行，而是应该返回正确的操作提示给用户。具体到本文的数据扩增系统，可靠性体现在：用户上传文件格式错误时，应当提示用户上传正确格式的司法裁判文书；用户未按要求填写信息时，应当提示用户信息填写错误。

安全性表示用户数据应当是安全的。用户通过系统扩增后的数据文件应当具备隐私性，即文件只能被其作者或系统管理员操作。从角色管理的角度来看，即用户只能看见自己扩增的司法文本，系统管理员可以看到系统内所有的司法文本。

容错性表示系统数据应当是可备份的。当存储用户数据的某台服务器发生故障时，系统不可以停止服务，此时系统应当可以从分布式集群中的其他服务器中拷贝备份数据，保障系统的持续运行。即系统不会因为一个节点数据的丢失，而导致数据的永久性丢失。

3.2.3　系统用例图

基于上面分析的系统功能性需求，数据质量驱动的司法文本数据扩增系统的用例图如图 3.1所示。

作为平台的使用者，用户可以根据需要上传本地司法文本数据，系统会根据用户上传的数据进行文本解析和数据扩增，数据扩增会先后经历主观质量维

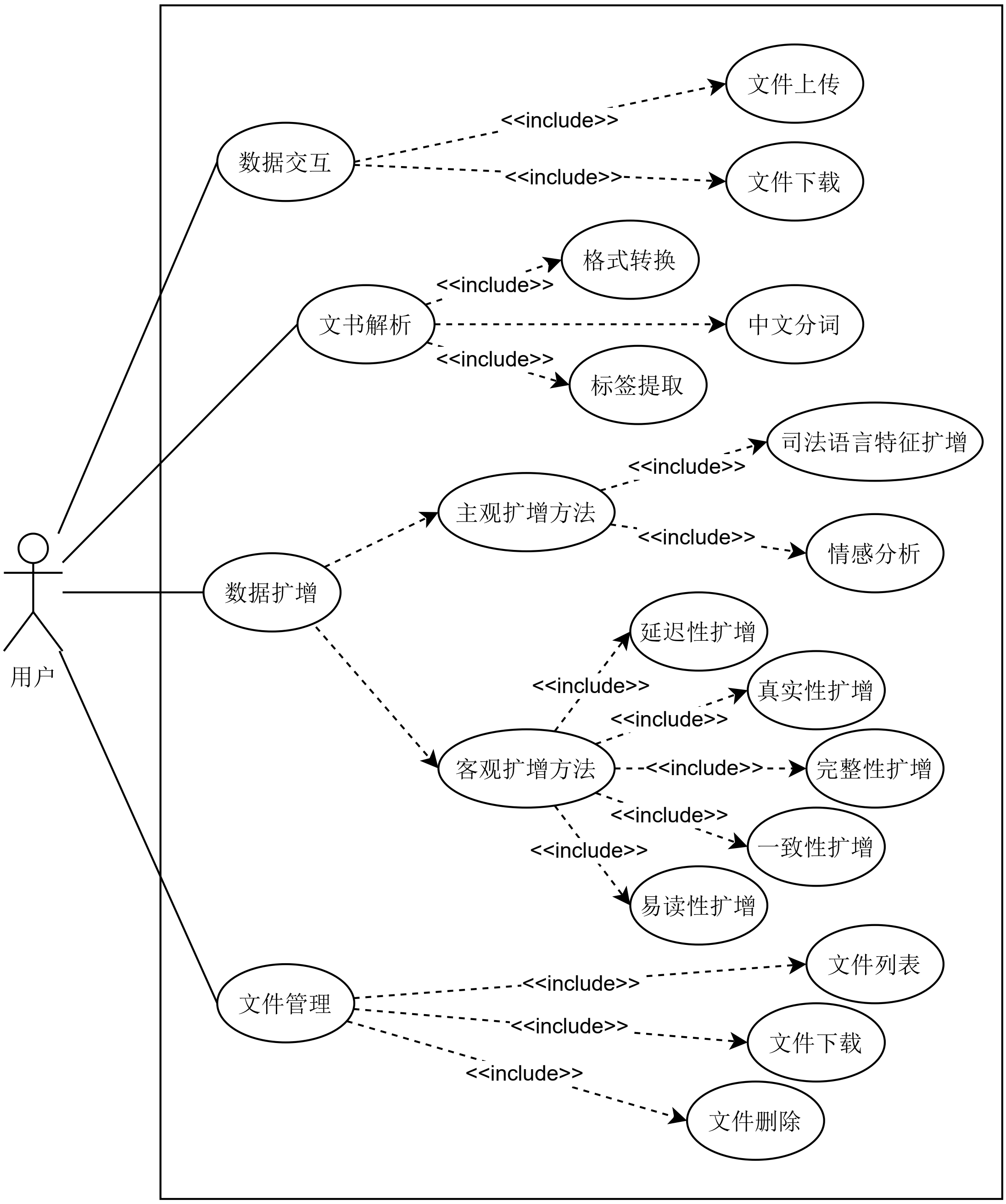

图 3.1: 系统用例图

度的扩增和客观质量维度的扩增，扩增后的司法文本会存储在 HDFS 中，用户可以查看并下载自己扩增过的司法文本。

3.2.4 系统用例描述

基于系统用例图，本小节将详细描述系统用例。数据质量驱动的司法文本数据扩增系统的用例包括司法文本数据上传、司法文本数据下载、司法文书解

析、主观质量维度的扩增、客观质量维度的扩增和司法文本文件管理。表 3.3为系统用例表，该表展示了用例和需求之间的关系。接下来将对系统用例进行详细描述。

表 3.3: 系统用例表

用例 ID	**用例名称**	**需求 ID**
UC1	司法文本数据上传	R1
UC2	司法文本数据下载	R2
UC3	司法文书解析	R3
UC4	主观质量维度的扩增	R4
UC5	客观质量维度的扩增	R5
UC6	司法文本文件管理	R6

司法文本数据上传用例描述如表 3.4所示。司法文本数据上传是数据扩增的前提，用户在数据交互模块完成文书的上传，才能开展后续步骤。用户上传的文件会存储到 HDFS 中，存储时系统会计算文件的 sha1 摘要，将其作为存储所用的文件名，此举是为了避免存储相同文件。文件的信息则存储在 MySQL 数据库中，其中包括用户文件名、系统文件名、上传时间等，这些信息的存储能够方便文件的查找。

表 3.4: 司法文本数据上传用例描述

ID	UC1
名称	司法文本数据上传
参与者	数据扩增系统用户
触发条件	用户选择文件上传
前置条件	1. 用户按照系统要求的格式准备好司法文本 2. 用户登录系统 3. 用户处于文件上传的页面
后置条件	文件上传成功或失败
优先级	高
正常流程	1. 用户按照文件格式要求上传司法文件 2. 系统计算文本 sha1 哈希值作为 HDFS 文件系统中的文件名，并将文件存至 HDFS
异常流程	1a. 用户上传文件格式非法 　1. 系统拒绝该请求 　2. 系统返回提示信息，用户重新上传 2a. 若 HDFS 内已有系统文件名相同的文件 　1. 不再重复上传至 HDFS

司法文本数据下载用例描述如表 3.5所示。用户在系统完成数据扩增后，可

以选择下载扩增后的司法文本。在下载时，系统会先从 MySQL 数据库中获取文件的系统文件名，然后再依据系统文件名将文件从 HDFS 中下载到用户计算机本地。

表 3.5: 司法文本数据下载用例描述

ID	UC2
名称	司法文本数据下载
参与者	数据扩增系统用户
触发条件	用户点击下载按钮
前置条件	1. 用户已在系统内扩增过至少一条文本 2. 用户处于文件下载页面
后置条件	文件下载成功
优先级	高
正常流程	1. 用户进入文件下载页面 2. 用户选择文件并点击下载按钮 3. 系统根据文件名从 HDFS 中下载文件至本地

司法文书解析用例描述如表 3.6所示。司法文书解析作为司法文书扩增的文本预处理模块，承担了把用户上传的司法文本转化为系统可操作格式数据的任务，具体步骤为：若上传的司法文书为文本格式，则将其转换为结构化的 json 数据格式，以便后续操作；用 jieba 对司法文书进行中文分词操作，支持后续数据扩增；提取文书内容标签，例如时间标签、人名标签等，作为后续扩增的基础。司法文书解析会把用户上传的文件转换为 json 格式的文件，然后将其存储到 HDFS 中。

表 3.6: 司法文书解析用例描述

ID	UC3
名称	司法文书解析
参与者	数据扩增系统用户
触发条件	用户上传文件成功后自动触发
前置条件	用户成功上传司法文本
后置条件	1. 系统完成司法文书的解析 2. 将解析后的 json 格式文件存入 HDFS
优先级	高
正常流程	1. 用户成功上传文件后，系统将文件从 HDFS 中导出，并转换成 json 格式 2. 用 jieba 对文本进行分词操作，存储在 json 文件里 3. 提取裁判文书内容标签，存储在 json 文件里

主观质量维度的扩增用例描述如表 3.7所示。主观质量维度的扩增是本系统

两大扩增模块之一，对整个系统而言非常重要。这一部分包括了司法语言特征扩增和基于文本情感分析的扩增。

司法语言特征扩增会学习司法裁判文书数据的语言特征，首先选取高质量司法裁判文书作为训练数据，用中文分词、词向量技术、机器学习技术来训练高质量裁判文书，得到 CBOW 模型，CBOW 模型会根据词向量相似度来替换待扩增文本的词语；基于文本情感分析的扩增则是为了保证司法文本保持情感中立性，这里会基于 Hownet 情感词库和部分标记情感特性的司法文本，训练融合得到一个情感词库，在进行词语替换时，避免使用带有主观情感偏向的词，就可以保证扩增后的文本具备情感中立性。

表 3.7: 主观质量维度的扩增用例描述

ID	UC4
名称	主观质量维度的扩增
参与者	数据扩增系统用户
触发条件	上传文件被转换成 json 文件后自动触发
前置条件	用户上传的司法文本已完成文书解析
后置条件	1. 系统完成主观质量维度的扩增 2. 系统将数据传入客观质量维度的扩增模块
优先级	高
正常流程	1. 系统从 HDFS 中载入已完成文书解析步骤的文件 2. 系统对司法文本进行主观质量维度的扩增 3. 系统将扩增后的数据传入客观质量维度的扩增流程

客观质量维度的扩增用例描述如表 3.8所示。客观质量维度的扩增也是本系统两大扩增模块之一，和主观质量维度的扩增模块一样，对整个系统来说至关重要。这一部分会先后对司法文本进行延迟性扩增、真实性扩增、完整性扩增、一致性扩增和易读性扩增。延迟性扩增保证案件发生时间和反应时间之间的时差不会过长，真实性扩增保证文书对案件的还原程度高，完整性扩增保证文书在完整程度上满足数据期望，一致性扩增保证文书的前后语言规范一致，易读性扩增保证文本语言的简明性。

司法文本文件管理用例描述如表 3.9所示。司法文本文件管理用于管理用户文件。用户登录进入文件列表页面后，可以查看用户文件名、文件大小、上传时间等文件信息，且可以不限制次数地下载扩增文本文件至本地计算机。至于文件的删除，系统针对可能存在的重复文件，增加了一个数据类型用于定义文件重复的次数。如果删除的文件不是重复文件，则直接在 HDFS 中删除这个文件；如果删除的文件是重复文件，则只修改该文件的重复次数，不从 HDFS 中将此文件删除。

表 3.8: 客观质量维度的扩增用例描述

ID	UC5
名称	客观质量维度的扩增
参与者	数据扩增系统用户
触发条件	完成主观质量维度的扩增后自动触发
前置条件	司法文本已完成主观质量维度的扩增
后置条件	1. 系统完成客观质量维度的扩增 2. 系统将数据存入 HDFS
优先级	高
正常流程	1. 系统获取主观质量维度的扩增结果 2. 系统进行客观质量维度的扩增 3. 系统将扩增后的数据存入 HDFS

表 3.9: 司法文本文件管理用例描述

ID	UC6
名称	司法文本文件管理
参与者	数据扩增系统用户
触发条件	用户点击进入文件列表
前置条件	用户进入文件列表页面
后置条件	若有文件，则删除成功
优先级	中
正常流程	1. 用户点击进入文件列表页面 2. 用户查看已扩增司法文本文件信息 3. 用户点击删除按钮，删除相应的司法文本数据
异常流程	2a. 用户未扩增过文本，文件列表为空 1. 用户看不到已扩增司法文本的文件信息 3a. 用户未扩增过文本，文件列表为空 1. 用户无法删除文件

3.3 系统总体设计

3.3.1 系统整体架构设计

数据质量驱动的司法文本数据扩增系统的整体架构图如图 3.2所示，系统包括前端、服务端、存储和分布式服务四大部分，其中服务端又分为文书解析模块、主观质量维度的扩增和客观质量维度的扩增三个模块。

前端部分设计了本系统的用户交互界面。网页前端页面的搭建采用了 Bootstrap 和 jQuery 框架，在 B/S 架构下，以浏览器为中间媒介，与系统使用者进行交互。这种设计、交互方式使用户的操作更加简便。在业务数据传入前端模板的

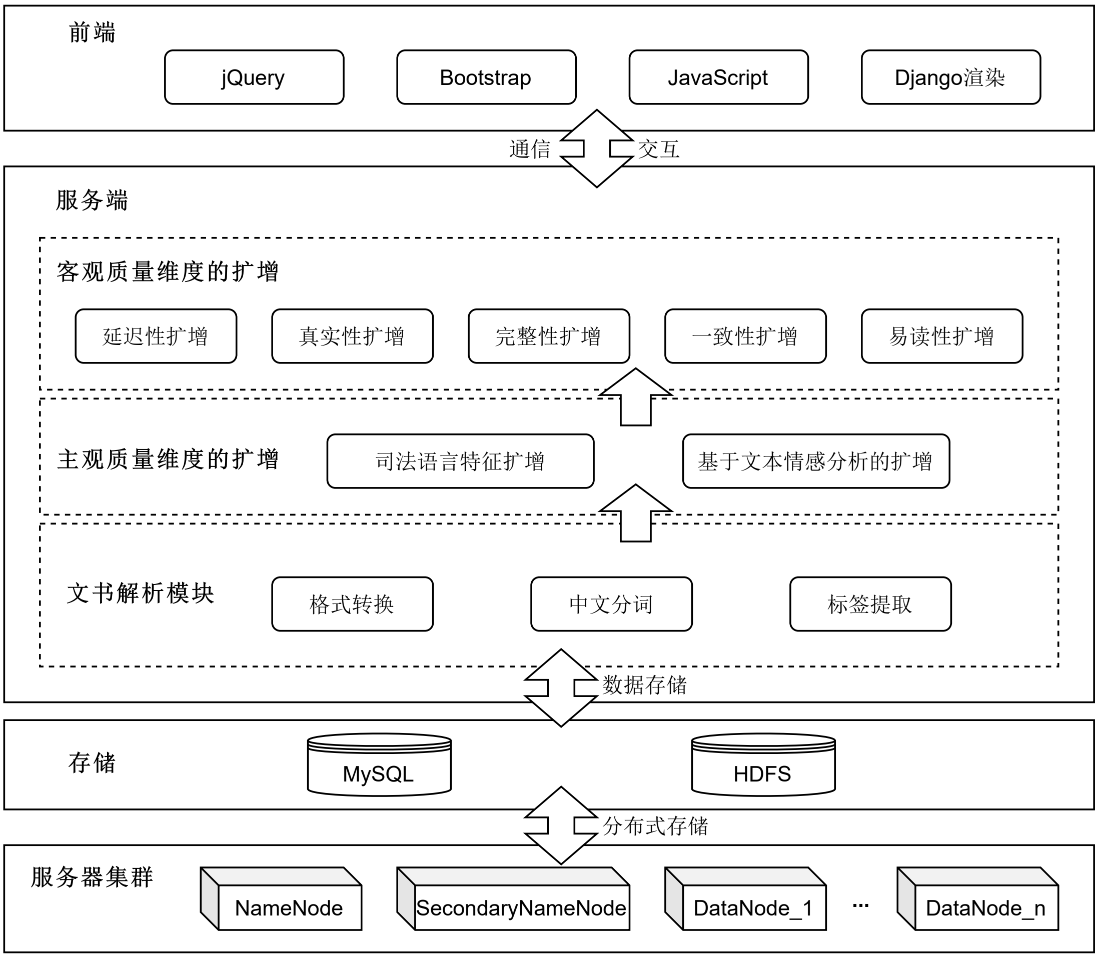

图 3.2: 系统整体架构图

过程中，系统采用了 Django 框架的模板引擎渲染功能。用户请求通过 https 协议的封装，从客户端传输到服务器端，待服务器端处理完成这个请求后，结果会自动返还给客户端。此外，文件的传输也是通过 https 协议。

服务端的客观质量维度的扩增模块属于系统的两大核心业务模块之一，主要负责司法文本在司法数据的客观质量维度层面的数据扩增。这一模块从主观质量维度的扩增模块获取数据，获取的数据是经过主观质量维度扩增过的 json 格式司法文本，json 格式的司法文本带有时间、人名、案由等标签。获取到数据后，本模块会基于带有标签的司法文本，进行延迟性扩增、真实性扩增、完整性扩增、一致性扩增和易读性扩增五种客观质量维度的扩增，然后将扩增的数据传入数据存储模块。

服务端的主观质量维度的扩增模块同属于系统的两大核心业务模块之一，主要负责司法文本在司法数据的主观质量维度层面的数据扩增。这一模块的数

据输入由文书解析模块完成，文书解析模块会将解析好的json格式数据传入。获取数据后，本模块会进行两种主观质量维度的扩增，分别是司法语言特征扩增和基于文本情感分析等扩增，扩增完成后的数据仍然存到json文件里。本模块会把json文件传入客观质量维度的扩增模块。

服务端的文书解析模块是存储部分和数据扩增部分的中间连接者，主要负责解析用户上传的司法文本文件，属于文本分析流程的数据预处理模块。该模块有三个步骤，首先将司法文本文件转换成json格式，然后对文本进行中文分词，最后提取文本的内容标签。该模块从HDFS中获取文件，将文件解析为扩增模块可处理的json格式后，传入主观质量维度的扩增模块。

存储部分包括了MySQL和HDFS，两者配合完成数据存储的工作。关系型数据库MySQL负责存储结构化数据，包括用户信息和文件信息。其中用户信息有用户名、用户密码和用户权限等，文件信息有用户文件名、系统文件名、文件上传时间、文件大小和文件所属用户等。分布式文件系统HDFS负责存储用户上传的原文件，以及数据扩增过程中产生的相关扩增文件。

本系统采用服务器集群存储数据。集群中的NameNode用来管理HDFS的原数据，当客户端发送读写请求时，NameNode会处理；集群中的DataNode则用来存储数据，当有指令送达，DataNode会执行读写数据块的操作。

3.3.2　4+1 视图

4+1视图是由系统逻辑视图、系统开发视图、系统进程视图、系统物理视图和用户场景共同组成的软件架构视图，如图3.3所示。如果只用一张图来描述整个系统的架构，那么难度会很大，且可能会因为包含的元素过多而令人难以理解，因此采用4+1视图这种结构清晰的形式，从不同的系统参与者视角来描述整个架构。不同的系统参与者只关心自己的视角，因此每个视图只描述当前视角面对的问题，这样可以使架构视图更容易被人理解。

用户场景描述了系统开发的目的，即用户需求的业务场景。本文已经在3.2用户需求分析中描述了用户场景。

逻辑视图从业务角度来介绍系统，它具体描述了系统提供的功能及服务，具体如图3.4所示。在逻辑视图中，HttpResponse实现了网络通信的基本功能，它是整个系统进行网络通信的基类。DataUpload和DataDownload都继承了HttpResponse类，分别负责司法文本文件的上传和下载功能。DataInteraction是负责数据交互的类，司法文本文件的存储和查找都依赖它，HDFS和MySQL也都连接在DataInteraction的接口上。其中HDFS为系统提供文件的读写服务，MySQL数据库则为系统提供数据库数据的读写与查询服务。FileAnalysis在系统中负责解

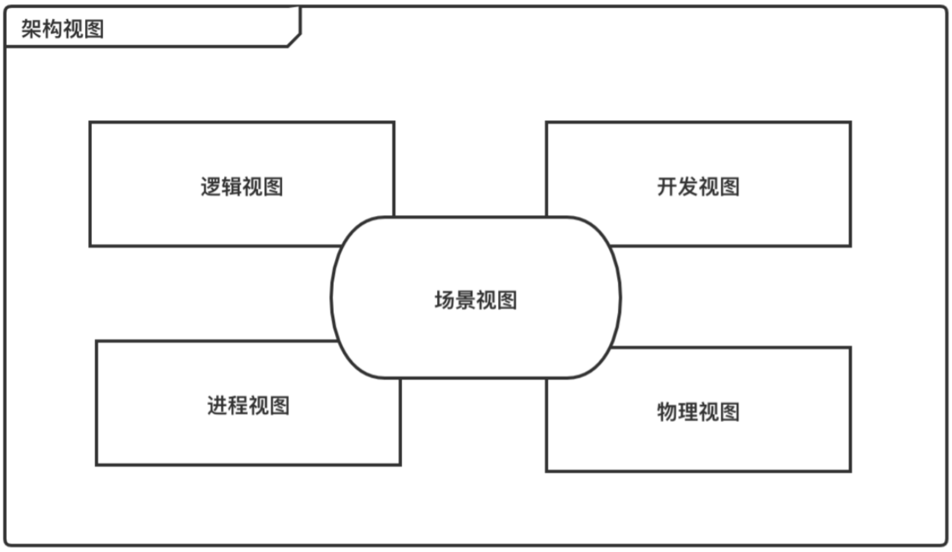

图 3.3: 系统架构视图

析用户上传的司法文书，包括了文书格式转换、中文分词和标签提取。DataAugmentation 是系统进行数据扩增的接口，它被 SubjectiveAugmentation 和 ObjectiveAugmentation 两个类实现。其中，SubjectiveAugmentation 是系统进行主观质量维度的扩增所使用的基类，ObjectiveAugmentation 则是系统进行客观质量维度的扩增所使用的基类。SubjectiveAugmentation 负责主观质量维度的扩增，其子类是 FeatureAugmentation 和 SentimentAnalysis，分别负责司法语言特征扩增和基于情感分析的扩增。ObjectiveAugmentation 负责客观质量维度的扩增，其子类有 LatencyAugmentation、AuthenticityAugmentation、IntegrityAugmentation、ConsistencyAugmentation 和 LegibilityAugmentation，分别负责延迟性扩增、真实性扩增、完整性扩增、一致性扩增和易读性扩增。

开发视图从开发者的角度来介绍系统，将系统的开发架构描绘出来，如图 3.5所示。开发视图分为三层，从上到下依次为 UI 层、Server Logic 层和 TechService 层。

UI 层由 jQuery、Bootstrap、JavaScript 和 Django 的交互组件共同构成。它负责用户交互界面的设计与搭建，用户通过 UI 层访问数据扩增服务和司法文书文件管理服务，也通过它传递用户文件和数据到服务器端。此外，数据交互模块也通过它展示扩增结果，也支持用户通过浏览器下载扩增结果文件。

Server Logic 层描述了系统逻辑功能的实现。该层由 DataInteractionService、DataManagementService、FileAnalysisService 和 DataAugmentationService 四部分组成。DataInteractionService 连接 HDFS 和 MySQL 两个存储介质，负责文件和

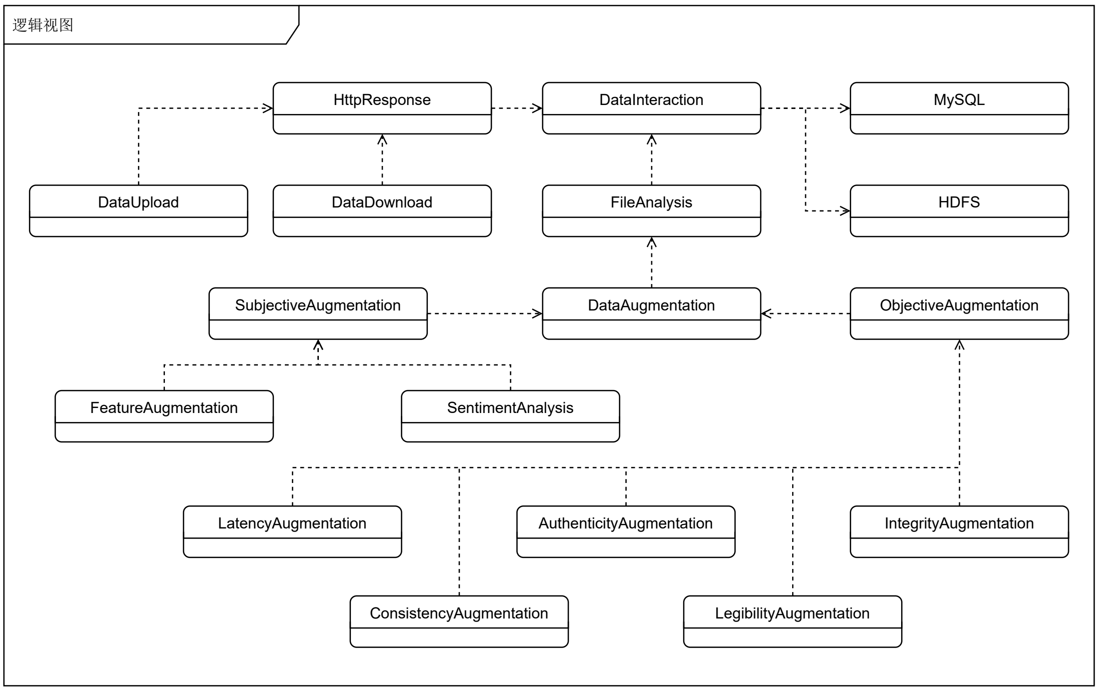

图 3.4: 系统逻辑视图

数据的交互，包括文件的存取和数据的读写。FileAnalysisService 负责分析文本，包括文件结构化、分词和标签提取。DataAugmentationService 则负责司法文本的扩增。

TechService 层包括了系统开发所用到的技术服务，这些技术服务为开发者提供了便捷直接的调用方法，让开发者可以直接使用这些方法来开发。本系统开发过程中用到的技术服务有 MySQL、HDFS 和服务器集群技术。

进程视图从系统集成人员的角度描述系统的数据通信过程，本系统的进程视图如图 3.6所示。该图可以有效地协助技术人员快速掌握系统的数据流动逻辑。在本系统中，进程的启动节点是用户上传原文件，待用户成功上传待扩增的司法文本后，DataUpload 会将原文件传至文书解析模块，然后 FileAnalysis 会进行文书解析，DataInteraction 将解析后的文书存储到 HDFS 中，同时也会解析出文件相关信息并存储到 MySQL 中。在进行数据扩增前，系统会把文件位置、系统文件名、文件在 MySQL 中的键值提取出来，传入数据扩增模块，这些将作为数据扩增对象的文书信息。DataAugmentation 负责数据的扩增，DataInteraction 则会将扩增后的司法文本存储到 HDFS，文件信息将会被存储在 MySQL。DataDownload 获取到司法文本的系统文件名和存储位置后，可以通过客户端浏览器将扩增后的文本文件下载到本地。GetFileInfo 负责将用户生成过的所有文本文件及其相关信息从数据库中读出，用于文件信息的展示与查看。

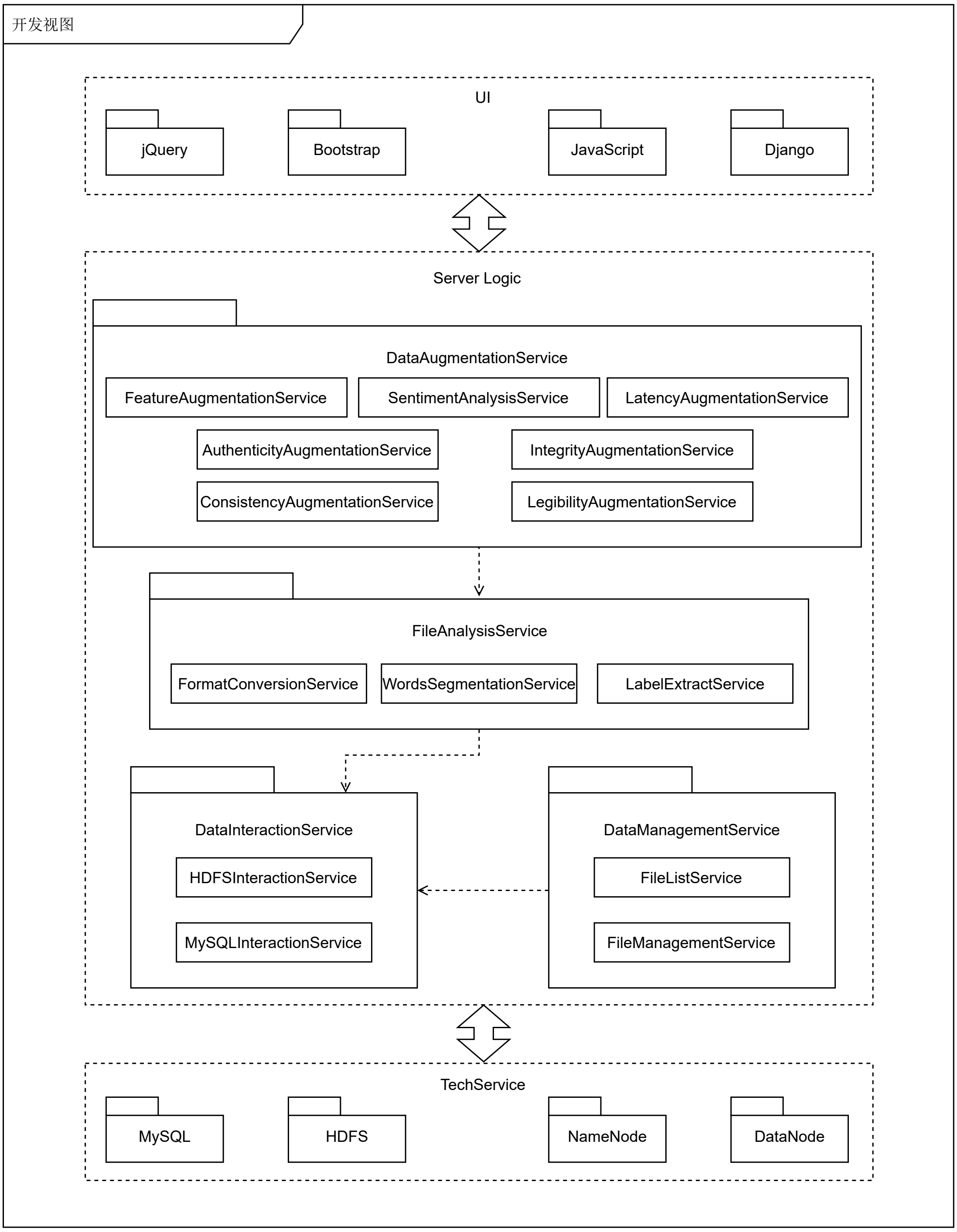

图 3.5: 系统开发视图

物理视图从系统运维人员的视角描述软件和硬件如何一一对应。它反映了系统在真实部署状态下的运行模式，让系统运维人员更快速地理解系统运行情况。如图 3.7所示，本系统提供的服务由用户主动发起。用户在本地主机浏览器

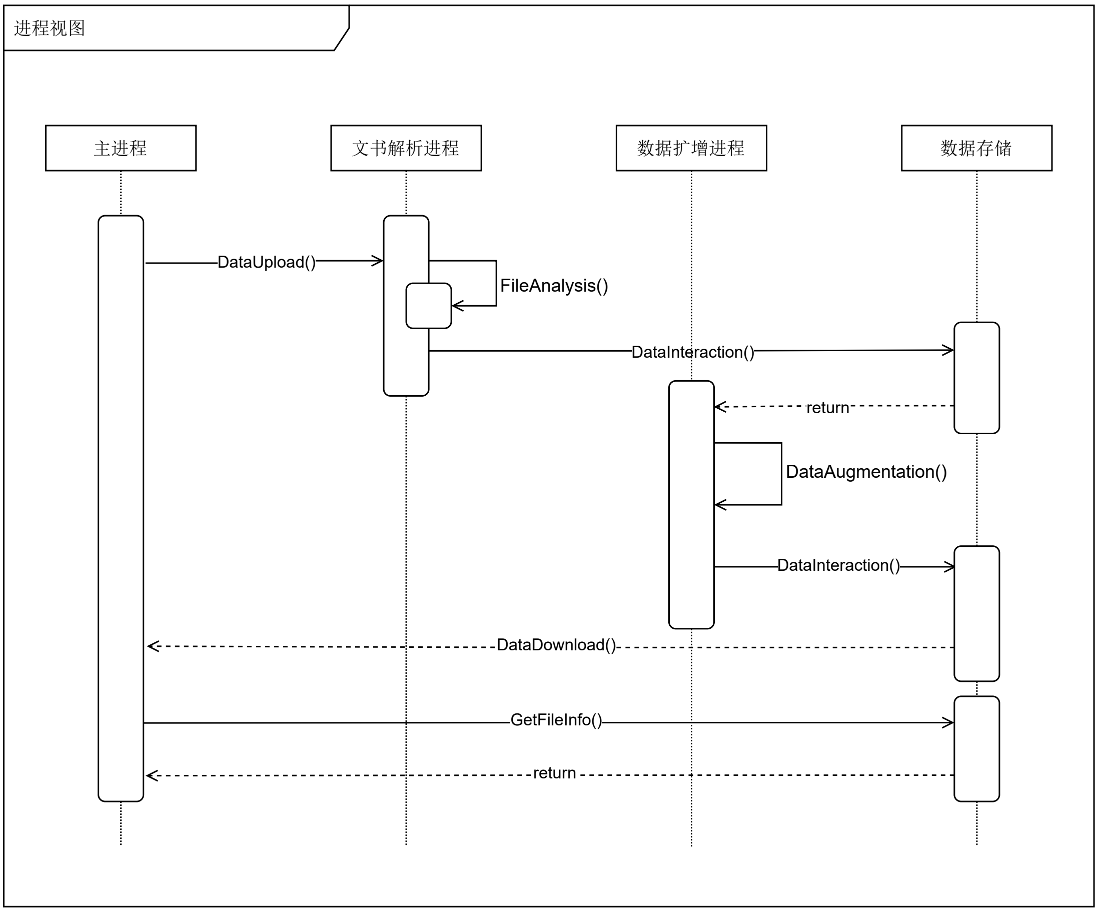

图 3.6: 系统进程视图

发出 Http 请求，申请访问数据扩增系统网站。Nginx 服务器带有代理转发功能，可以应对跨域的问题，让用户的请求到达应用服务器。本系统的应用服务器是基于 Django 框架搭建的，Django 的 WSGI 服务器可以解析用户发送的 Http 请求报文，它会把 URL 映射到视图函数中，再利用视图函数处理系统业务，最后再返回 Http 响应报文。MySQL 数据库部署在数据服务器上，用于数据的存储和查询，数据扩增系统通过 TCP 传输协议与数据库服务器进行通信。HDFS 被部署在服务器集群上，它负责系统文件的存储，系统通过 SSH 传输协议与 HDFS 进行通信。

3.3.3　系统基本功能模块

根据以上系统架构分析和 4+1 视图，本系统的设计将分为数据交互模块、数据管理模块、文书解析模块和数据扩增模块四部分，具体如图 3.8所示。

四个模块的具体拆解如下：数据交互模块分为了数据上传、数据存储和数据下载三部分；数据管理模块由文件查询和文件管理组成；文书解析模块有文

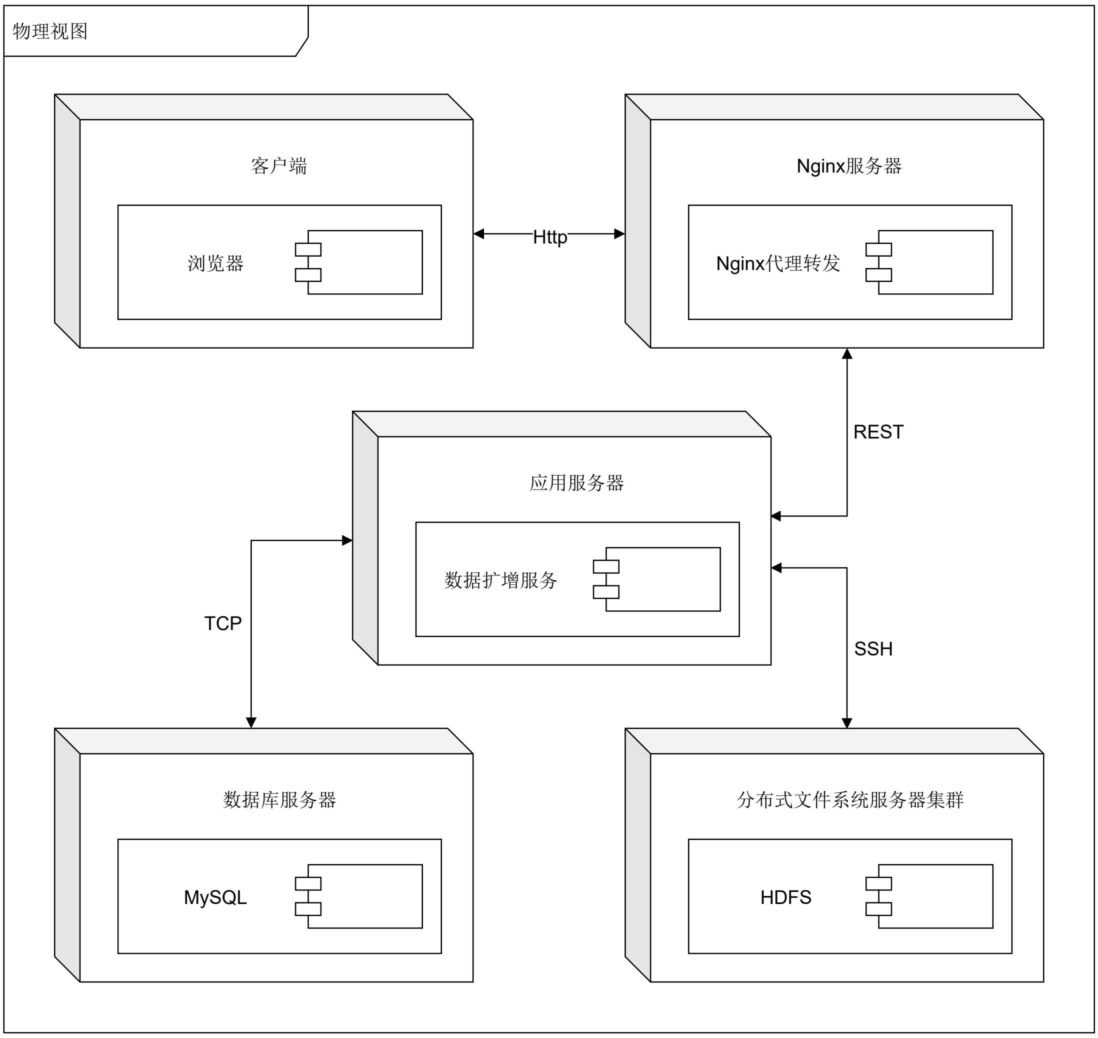

图 3.7: 系统物理视图

书格式转换、文本中文分词和标签提取三部分；数据扩增模块则分为了司法语言特征扩增、基于文本情感分析的扩增、延迟性扩增、真实性扩增、完整性扩增、一致性扩增和易读性扩增。各个模块的具体设计将在本章后面的部分详细介绍。

3.4　数据交互模块设计

数据交互模块承担了连接用户、系统和数据库的角色，它负责客户端到服务端再到存储端的数据交互工作。该模块的具体功能为上传用户的待扩增原文件并存储到 HDFS、根据用户指令查询数据、传输扩增数据到前端展示、下载扩增文本到客户端。

数据交互模块的核心类图如图 3.9所示。模块内的类和接口有 HttpResponse、DataUpload、DataDownload、DataTransfer、DataInteraction、DataInteractionHDFS、

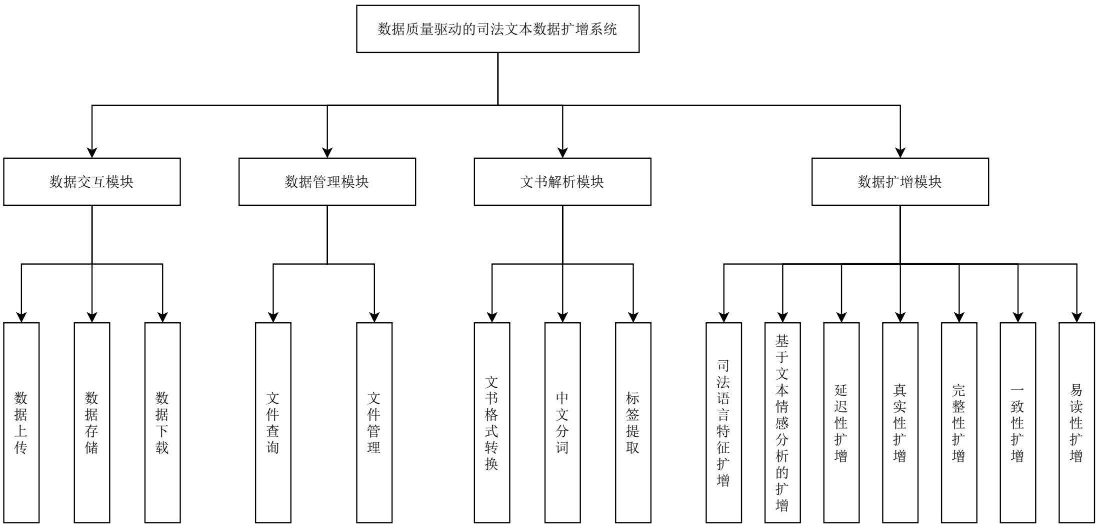

图 3.8: 系统基本功能模块图

DataInteractionMySQL。其中负责网络传输的基类是 HttpResponse，它会实现系统所需通信服务的基本功能。DataUpload 类继承了 HttpResponse 基类，它承担了待扩增文本上传的任务。和 DataUpload 类似，DataDownload 也继承了 HttpResponse，它承担了扩增文本下载的任务。DataTransfer 类负责将客户端的数据传输给 MySQL 数据库和 HDFS。DataInteraction 是系统连接 MySQL、HDFS 的接口，它由 DataInteractionMySQL 和 DataInteractionHDFS 类来实现。DataInteractionMySQL 将系统与 MySQL 数据库相连，并提供数据查询和读写的函数。DataInteractionHDFS 将系统与 HDFS 相连接，并提供文件存储、文件读写和文件下载等函数。

在实际应用中，存在同个文件上传多次的情况，为了避免同个文件的重复存储，减少存储资源浪费，本系统在存储文件时引入了信息摘要算法。该算法使用密码学加密技术，对数据进行计算，计算得到特征码。特征码具备高度离散且不可逆的特点，再加上特征码长度较长，所以两段不同文本的特征码不可能相同。

利用特征码的这一特性，本系统会将上传文件的文本内容进行加密计算，得到摘要特征码，然后以此为系统文件名存储该文件。如果用户又上传了一份内容相同的司法文本，那么这个文本会和已有文本具有相同的系统文件名，一旦系统判别出已有相同内容的司法文本，新传入的文件将不再被重复存储。系统在 MySQL 数据库中创建了数据表 FileNumber，该数据表用来记录某个系统文件名对应的文件数量，如果某个相同内容的司法文本被上传了两次，那么该文件的系统文件名在 FileNumber 表中对应的次数就为 2。当用户进行删除文件的

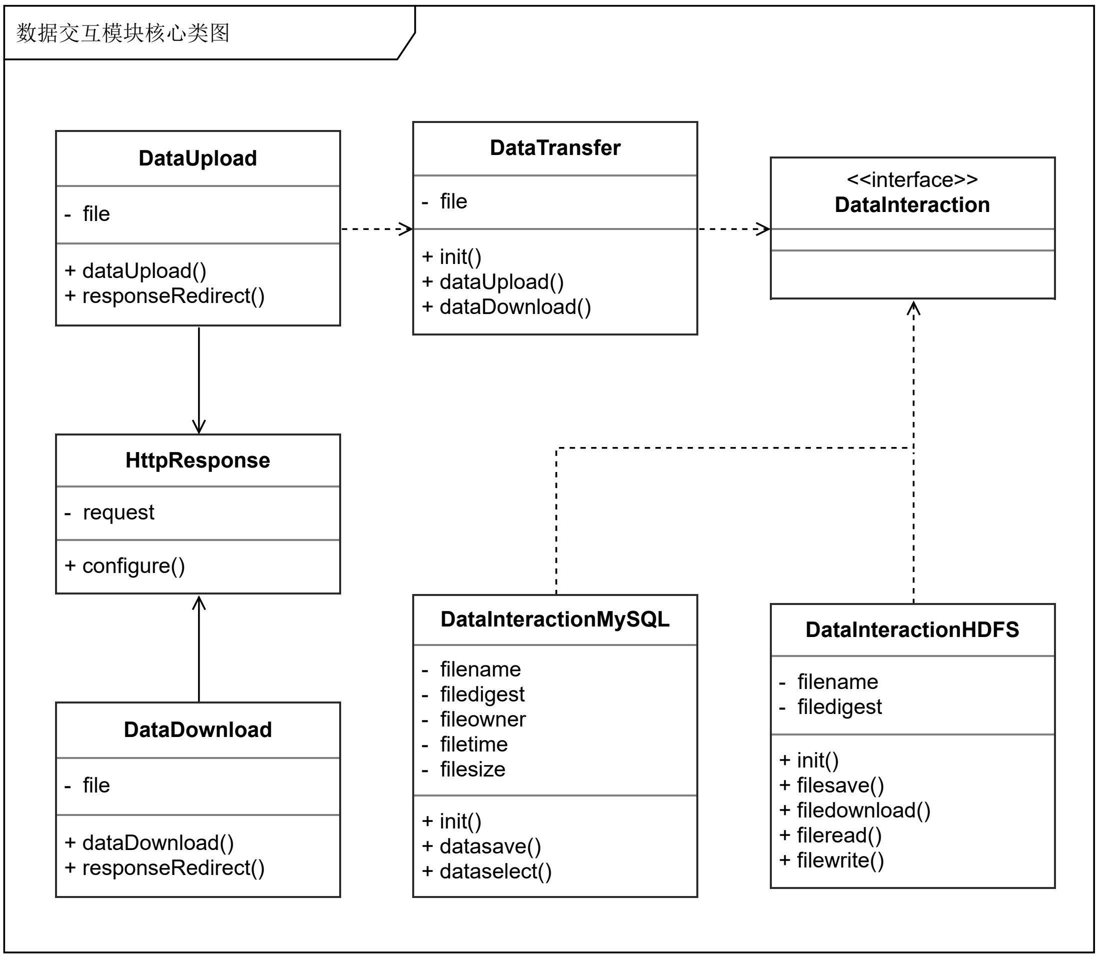

图 3.9: 数据交互模块核心类图

操作时，系统会检查 FileNumber 中链接的数量，如果数量大于 1，则不直接删除文件，只需要把链接数量减 1；若连接数量等于 1，则直接删除用户指定的文件。

系统文件删除流程如图 3.10所示。当用户选择删除某个文件时，系统需要先查询该文件内容所存储的信息表 File，以获取该文件的系统文件名 SysFileName，查询到后，再到 FileNumber 表中检索摘要特征码 digest 等于该 SysFileName 的链接数量。如果链接数量等于 1，则直接在 HDFS 中删除该文件；如果链接数量大于 1，则只要将该数字减 1，不删除 HDFS 中的文件。

系统文件存储流程如图 3.11所示。用户上传待扩增的司法文本后，系统会基于文本内容计算其摘要特征码 digest。该特征码作为 HDFS 中的系统文件名被保存，也就是说该文件在分布式文件系统内的文件名为 digest。在 MySQL 数据库中，系统会把包括用户文件名在内的所有文件信息存储到数据表 File 中。

当新文件被上传，系统会在 File 中检索是否有已存在的相同 digest 值的文件，

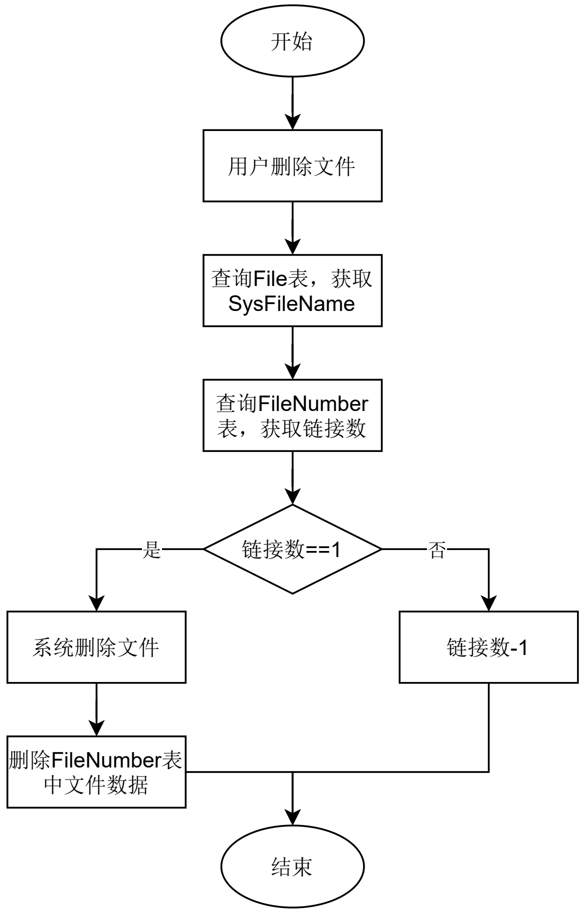

图 3.10: 系统文件删除流程图

如未检索到，则把文件以计算出来的 digest 值作为系统文件名存储到 HDFS 中，并将文件以用户文件名存储到 MySQL 数据库中，然后再到系统链接表 FileNumber 中添加系统文件名字段为 digest 的一行新数据，并把链接数这个字段置 1，表示目前该内容的司法文本数量为 1；如果新上传的文本是重复的，那么计算出来的 digest 也就是数据库中已有的，那么该文件不会被重复上传到 HDFS 中，而是直接将文件相关信息存储到 File 表内，并在系统链接表 FileNumber 中将该文件的链接数加 1，表示相同文件内容的文件数量增加了 1。

3.5 数据管理模块设计

数据管理模块主要负责用户在数据扩增系统中的扩增文件的管理工作，其活动图如图 3.12所示。用户首先进入数据管理页面，进入后，系统会在 MySQL 中检所该用户是否有已扩增且未被删除的文件。如果存在这种文件，则文件列表会展示所有已扩增文件的相关信息，包括用户文件名、文件扩增时间和文件

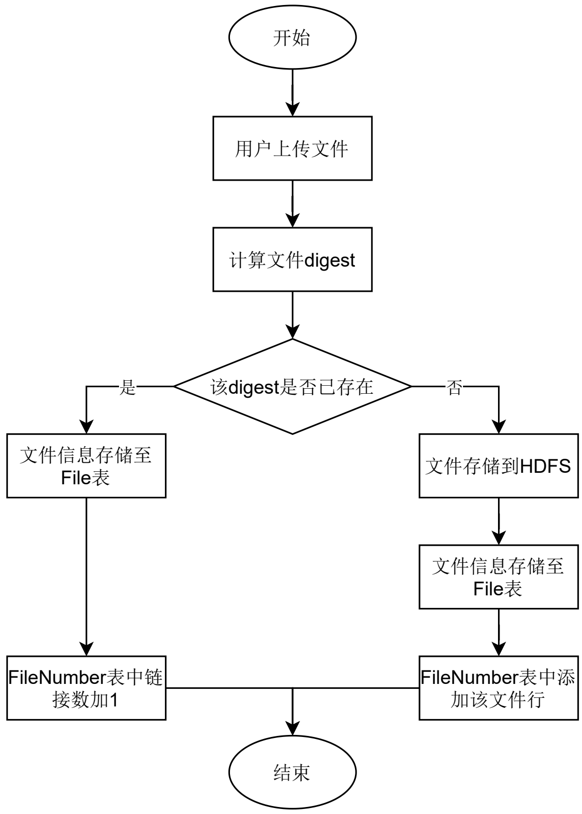

图 3.11: 系统文件存储流程图

大小等；如果不存在这种文件，那么文件列表则为空。

对于文件列表中的每一个文件，系统前端都提供了下载与删除的按钮。如果用户点击下载选项，系统会先到 MySQL 数据库中查询该文件的系统文件名，然后再到 HDFS 中根据系统文件名检索文件，最后再下载该文件到客户端。如果用户点击删除选项，系统会先到 MySQL 数据库中查询文件信息，并到 FileNumber 表中检查该文件是否是重复文件，如果不是重复文件，则直接删除该文件在所有存储介质中的信息，如果是重复文件，则把该文件在 FileNumber 表中的链接次数减 1，并且删除该文件在 File 表中的信息。

3.6　文书解析模块设计

文书解析模块负责对用户上传的司法文本进行文本解析，这一步也可以认为是数据扩增的预处理操作。

文书解析模块的核心类图如图 3.13所示。模块内的类有 FileAnalysis、FormatConversion、WordsSegmentation、LabelExtract，其中，FileAnalysis 是本模块所有类的基类，它作为数据预处理的基础，定义了输入和输出的文件名，同时也

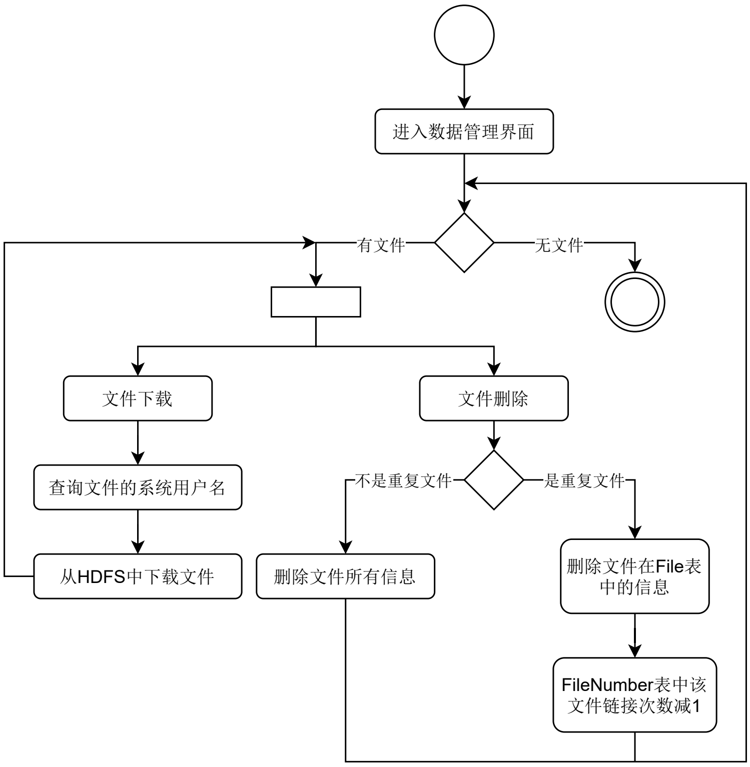

图 3.12: 数据管理活动图

定义了文件读写、文件存储到 HDFS、文件信息数据写入 MySQL 等一系列函数。FormatConversion、WordsSegmentation 和 LabelExtract 类是 FileAnalysis 基类的子类。FormatConversion 类定义了 format 函数用于文件格式的转换，WordsSegmentation 类定义了 segment 函数用于中文文本的分词，LabelExtract 类定义了 extract 函数用于文本内容标签的提取。

文书解析模块的流程图如图 3.14所示。用户先上传待扩增的司法文本文件，待用户上传后，系统检查上传文件格式是否为 json 或 txt 格式，如果不符合限定的格式，则会返回提示，要求用户重新上传。当用户按照格式要求上传司法文本文件后，系统就会执行文书解析操作。

文书解析的第一步是格式转换，系统会创建一个 json 格式的文件，用于存储解析后的数据，包括原文本内容、分词后的结果和内容标签等，这些内容会以结构化的形式存储在 json 文件中，将用户上传文件转换为规定格式的 json 文件就是格式转换。格式转换完成后，系统会对司法文本进行分词操作，分词使用 jieba，分词结果保存在 json 文件内。分词后会进行标签提取工作，司法文本内的时间标签、地名标签、案由标签等内容标签会被提取出来，并保存到 json 文件中。

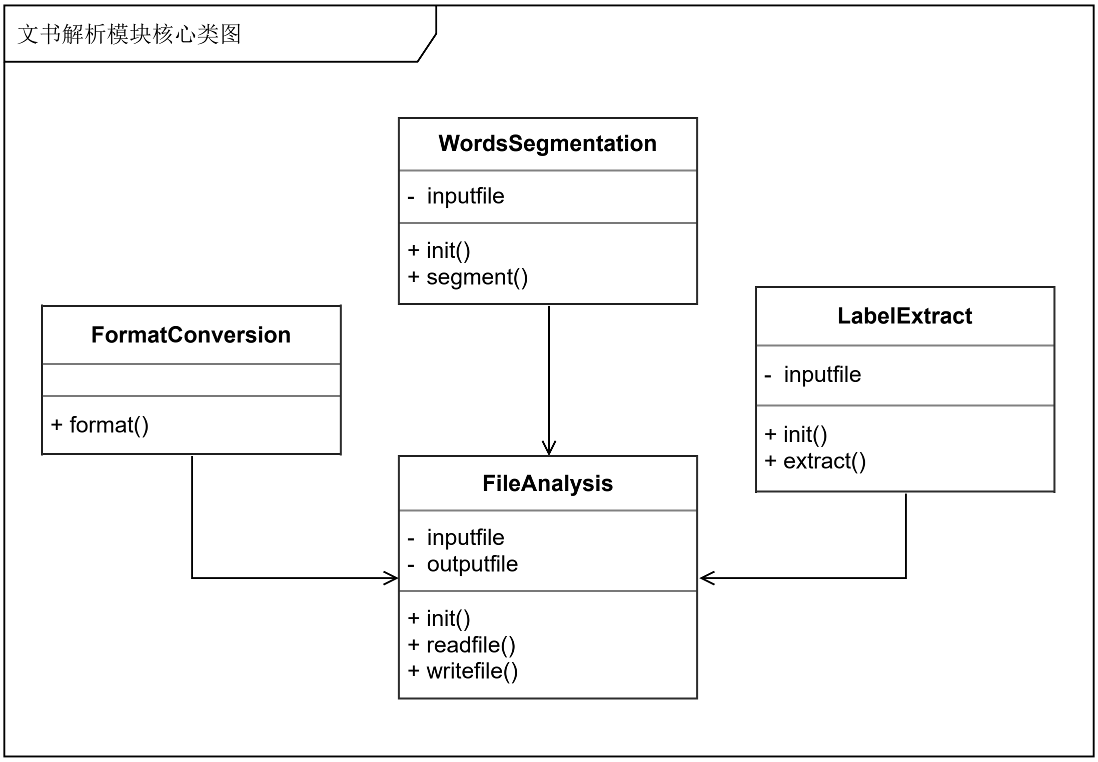

图 3.13: 文书解析模块核心类图

标签提取完成后，原文件也就被转换成了 json 文件。系统会把 json 文件的相关信息存储到 MySQL 数据库中，这里写入的内容有用户文件名、系统文件名、文件所有者、文件大小和上传时间等。至于文件本身的存储，将会根据系统文件名和定义的存储位置把文件存储到 HDFS 中。

文书解析模块将原文件解析成数据扩增模块可直接使用的数据，当系统进行数据扩增操作时，可直接从 HDFS 中导入解析过的 json 文件，不用重复进行数据扩增前的文本预处理工作，提高了数据扩增的效率。

3.7　数据扩增模块设计

数据扩增模块是整个系统的核心，是数据扩增功能技术实现的部分。本系统的数据扩增模块功能图如图 3.15所示。整个扩增模块分为主观质量维度的扩增和客观质量维度的扩增两部分。其中主观质量维度的扩增又分为司法语言特征扩增和基于文本情感分析的扩增；客观质量维度的扩增分为延迟性扩增、真实性扩增、完整性扩增、一致性扩增和易读性扩增。

数据扩增模块的核心类图如图 3.16所示。图中 DataAugmentation 为该模块的接口，该接口定义了数据扩增流程中会用到的一些通用接口，例如从 HDFS 中

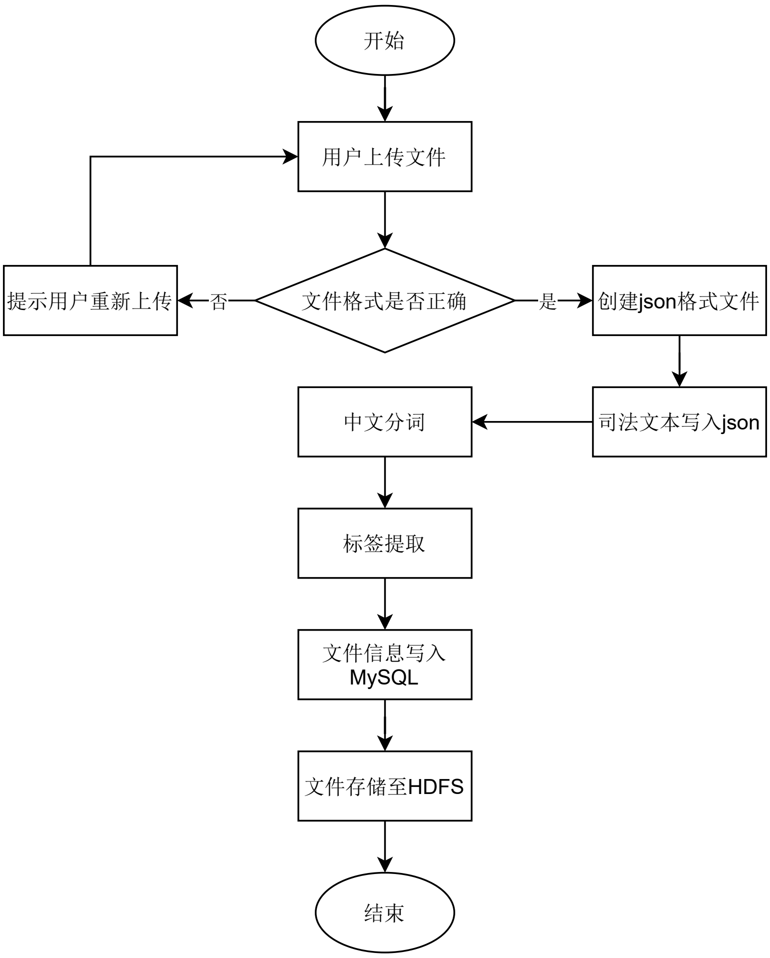

图 3.14: 文书解析模块流程图

存取文件、数据读写等。SubjectiveAugmentation 和 ObjectiveAugmentation 类实现了 DataAugmentation 接口，两者分别是主观维度扩增和客观维度扩增的基类，它们定义了数据扩增过程要用到的基础函数。

FeatureAugmentation 和 SentimentAnalysis 类继承自 SubjectiveAugmentation 类。FeatureAugmentation 的功能是对司法文本进行司法语言特征扩增，featureaugmentation() 方法基于 CBOW 模型进行词替换。SentimentAnalysis 的功能是对司法文本进行基于文本情感分析的扩增，sentimentanalysis() 方法基于情感词典控制替换词的情感中立性。

LatencyAugmentation、AuthenticityAugmentation、IntegrityAugmentation、ConsistencyAugmentation 和 LegibilityAugmentation 类继承自 ObjectiveAugmentation 类。五个类分别负责司法文本的延迟性扩增、真实性扩增、完整性扩增、一致性扩增和易读性扩增，客观性司法词库是进行扩增时用到的词替换库，latencyaugmen-

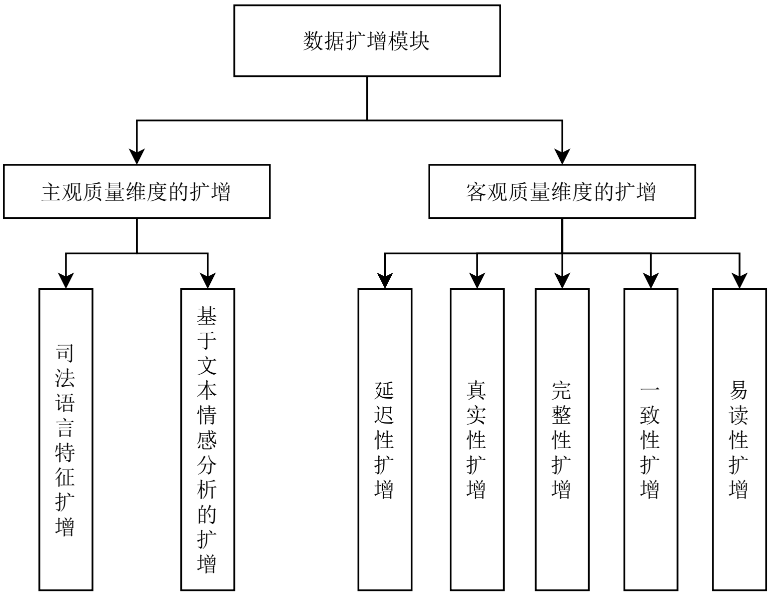

图 3.15: 数据扩增模块功能图

tation()、authenticityaugmentation()、integrityaugmentation()、consistencyaugmentation() 和 legibilityaugmentation() 分别是五个类中定义的相应的数据扩增方法。

数据扩增模块的流程如图 3.17所示。在对待扩增文本操作前，需要先训练出 CBOW 词向量模型，并生成司法文本情感词库。对于 CBOW 词向量模型的训练，系统会先对一批高质量的司法裁判文书进行分词，然后用 word2vec 中的 CBOW 模型训练出词向量。对于司法文本情感词库，系统会对一批标注了情感倾向的司法文本进行训练，并把训练结果和 Hownet 词库融合，得到用于司法领域的司法文本情感词库。

在得到 CBOW 词向量模型和司法文本情感词库后，数据扩增模块会读取经过文本分析的 json 司法文本文件，进行主观质量维度的扩增。首先是司法语言特征扩增，该扩增方法学习了大量司法裁判文书的文本特征，将司法领域的文字特点用机器学习技术浓缩在了一个 CBOW 词向量模型中。系统将待扩增文本作为输入，放进训练好的 CBOW 模型，可以得到经过词替换的输出文本，这里词替换的方法是基于词向量相似性计算出来的，这个相似性就代表了两个词在司法领域下的语义相似度。为了避免司法语言特征扩增的结果存在情感倾向，所以对于 CBOW 模型训练出的词向量，要对其进行去情感化操作，这个去情感化过程就是基于文本情感分析的扩增过程。这个过程会把上面合成的司法文本情感词库中的非中立词语设为禁用词，在 CBOW 词向量模型中避免使用这些带有情感倾向的词汇，这样就保证了主观质量维度的扩增结果具备司法文书的情感中立性。

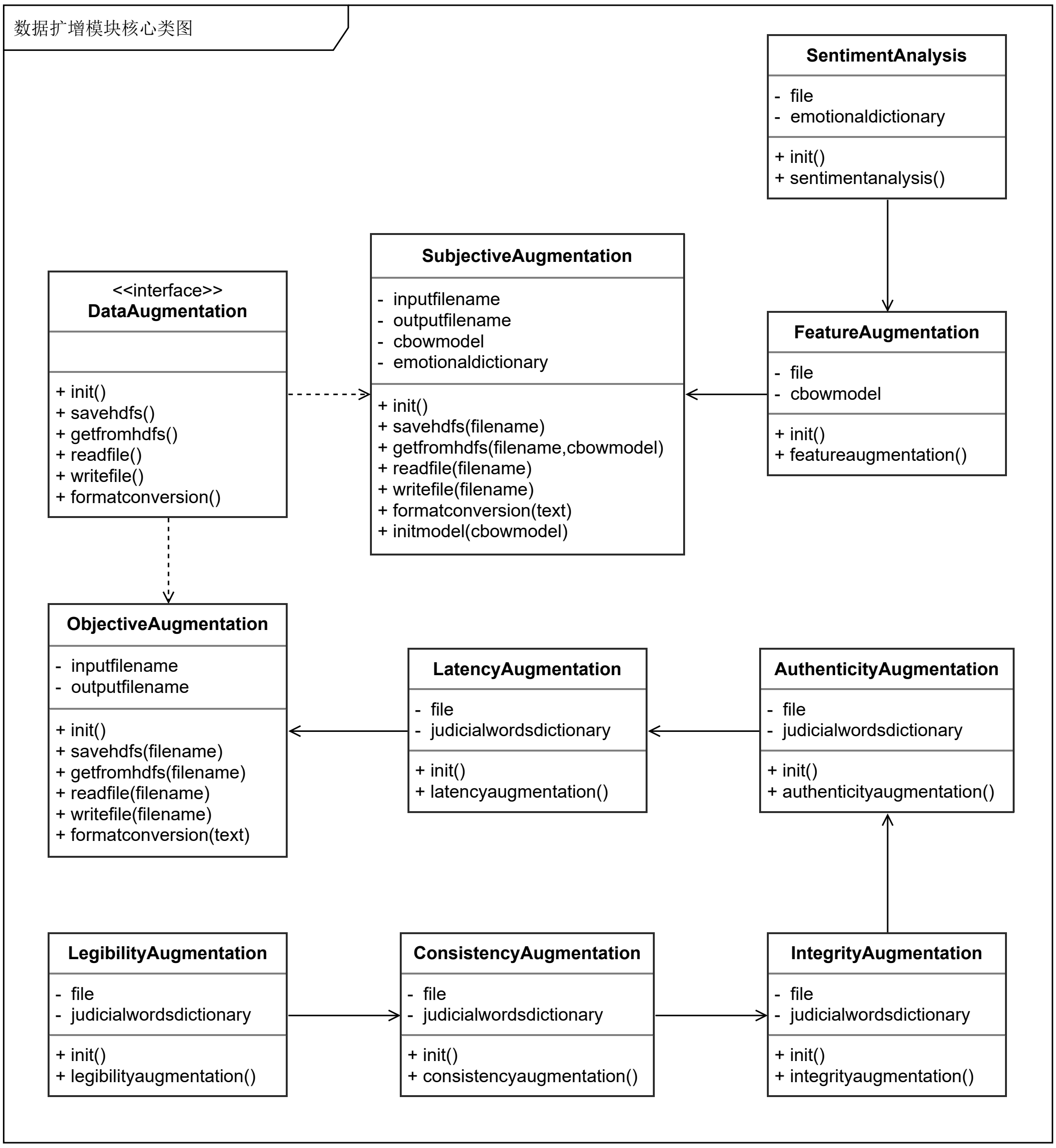

图 3.16: 数据扩增模块核心类图

在主观质量维度的扩增完成后，会进行客观质量维度的扩增。客观质量维度是基于客观信息论总结出来的一系列数据质量维度，针对这些维度进行扩增，可以保证扩增出来的结果具备较高的质量。延迟性扩增是第一个客观质量维度的扩增，裁判文书中会有很多时间出现，例如案发时间、立案时间等，这些时间的间隔应当在一个合理的区间内，如果时间间隔过长或者过短，都会与现实不符，导致数据质量低。因此在扩增过程中，系统会把带有时间标签的词提取出来，根据时间间隔的合理区间来扩增新的时间数据；真实性扩增从案由真实性和裁判依据真实性两个角度进行扩增，案由真实性保证司法文本的扩增案由都

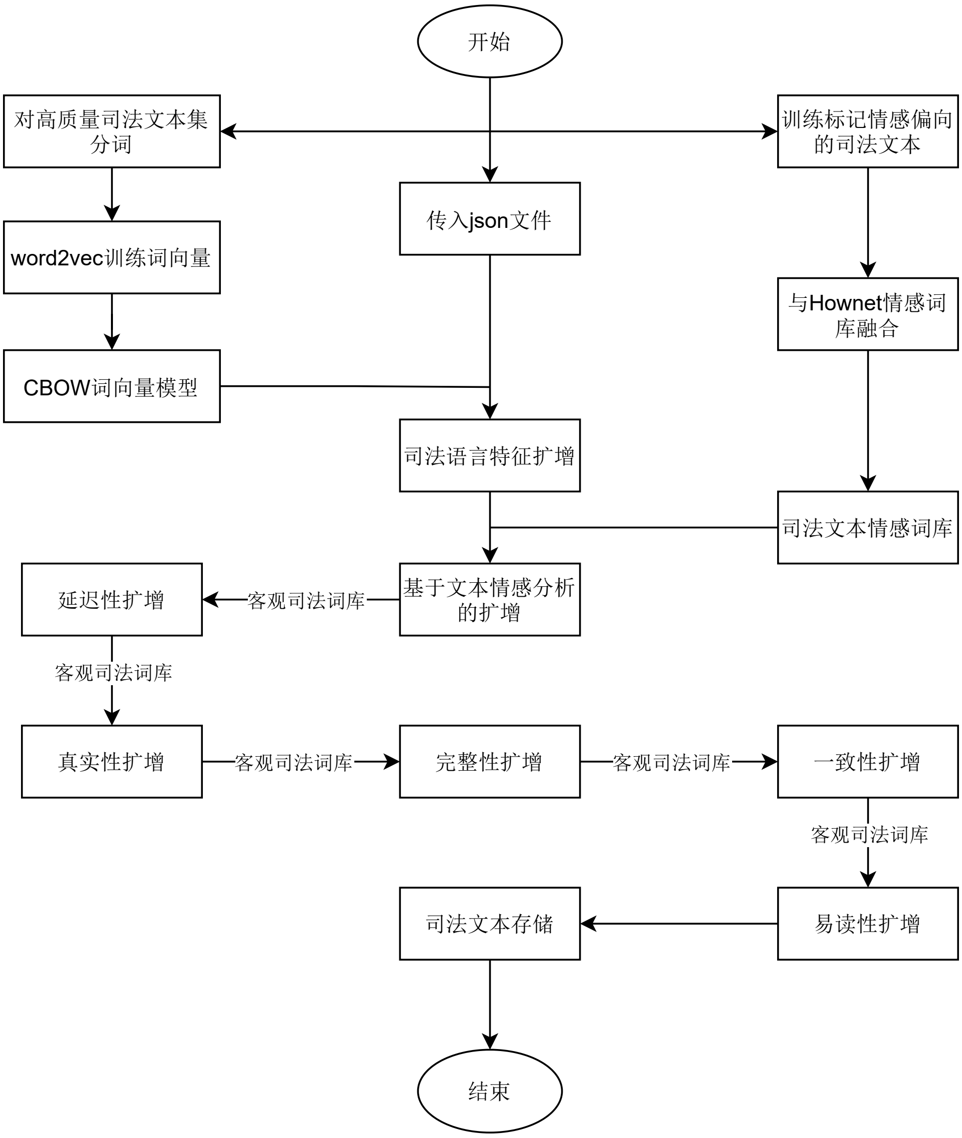

图 3.17: 数据扩增模块流程图

是在最高人民法院发布的案由序列内的，裁判依据真实性则保证扩增文本的裁判引用必须是人民法院规定的程序法和实体法条文；完整性扩增会在构建客观司法词库时加入参诉人、诉费承担等标签的文本内容，在扩增过程中对文本进行参诉人、诉费承担等标签的扩增，保证扩增文本完整地包含这些司法裁判文书的必要组成部分；一致性扩增将客观司法词库中的数字和标点符号统一成规定格式。当数字是文书对判决结果或分析过程的分项阐述、文尾的落款时间时，使用汉字数字，如“一、二、三”，此外都用阿拉伯数字。标点符号则根据《诉讼文书样式》中相应的规定进行词库的构建和数据的扩增；易读性扩增在词替换过程中约束替换规则，保证扩增过程是一词换一词，且替换词的长度不可以

超过原词长度的两倍。这种约束可以确保扩增后的文本长度不会太长、复杂性不会太高，对于司法文本阅读者来说比较友好。

完成上述数据扩增流程后，司法文书会以文本格式存储到文件系统中，并为用户提供下载接口。

3.8 本章小结

本章首先概述了系统的整体架构，在明确了系统开发目标的前提下，对系统进行了功能性需求分析和非功能性需求分析，并给出了系统用例图和用例描述。基于需求分析，本章在设计上把系统分为了数据交互模块、数据管理模块、文书解析模块和数据扩增模块四个部分，通过 4+1 视图从不同的视角介绍了系统，并详细介绍了四个模块的流程设计。

第四章　系统实现与实验测试

本章将会详细描述系统的技术实现，具体会分为数据交互、数据管理、文书解析和数据扩增四个部分来介绍。对于本文设计的数据扩增系统，本章还会展示系统的数据扩增结果，并对系统核心功能进行实验和测试。

4.1 数据交互实现

数据交互实现了系统使用者与系统之间数据的交互，它将用户、系统和数据库连成了一个整体，保障了用户数据和系统数据的正常传输。本系统的数据交互模块主要负责待扩增文件的上传、文件存储、数据查询、数据展示和文件下载等工作。具体的数据交互实现逻辑视图如图 4.1所示。

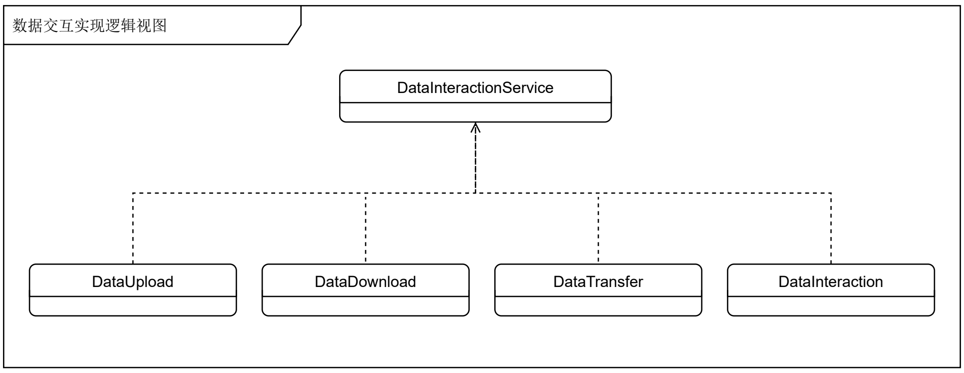

图 4.1: 数据交互实现逻辑视图

DataUpload 组件主要负责数据的上传任务。DataUpload 先从用户客户端发送的 HttpRequest 请求中解析出用户上传的原文件，然后检查文件格式是否为 txt 或者 json，如果符合系统规定的文件上传格式，就把文件和文件相关信息传给 DataTransfer 组件。之后会把文件存储到 HDFS，将文件信息存储到 MySQL。

DataDownload 组件主要负责数据的下载任务。当客户端请求下载某个文件时，DataDownload 会获取在数据库中该文件对应的键值，然后将键值传入 DataTransfer，接下来从 MySQL 中读取该文件的系统文件名和存储位置，并把文件从 HDFS 下载到用户本地。

DataTransfer 组件主要负责数据的传输任务。它将用户文件和系统数据作为传输对象，将其传递给系统的数据存储介质，包括把文件存储至 HDFS，以及把用户和文件的信息存储至 MySQL。此外，在系统完成数据扩增后的数据展示环

节，DataTransfer 会从 HDFS 中获取文本文件，并将其传输至 UI 界面，前端会将扩增文本展示在交互界面，具体的数据展示界面如图 4.2所示。

图 4.2: 数据展示界面

DataInteraction 组件是扩增系统与数据存储介质之间的交互接口。它负责系统在 MySQL 中的数据读写和数据查询功能，也负责系统在 HDFS 中的文件存取和文件下载功能。

对于可能遇到的较大文件的交互，以及大量文件的存储，本系统选用了 Hadoop 的核心组件 HDFS 作为文件系统。HDFS 为系统文件提供了分布式的存储环境，提高了系统文件存储空间的可扩展性，也确保了系统数据的安全性。

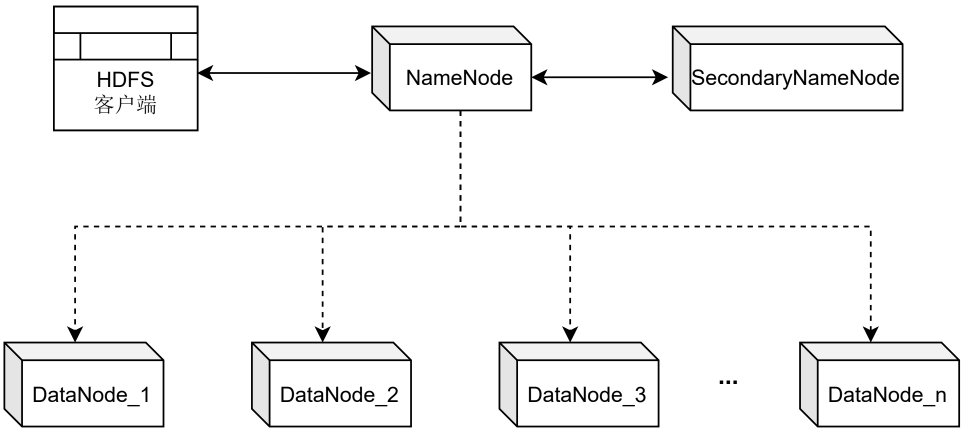

图 4.3: HDFS 架构图

HDFS 的架构如图 4.3所示。本系统是 HDFS 的主从结构，主要有三类角色，分别是 NameNode、Secondary NameNode 和 DataNode。其中 NameNode 负责与 HDFS 客户端进行读写请求的交互，并把文件的存储位置告知客户端，也负责管理块、副本策略等内容；Secondary NameNode 负责数据的合并，在合并后将数据传输给 NameNode 进行下一步处理，它的角色是 NameNode 的辅助；DataNode

是 HDFS 中的执行角色，它负责执行来自于 NameNode 的读写指令，将文件数据以分布式的方式存储到不同的块，并从不同的块读取数据。HDFS 配置的关键代码如图 4.4所示。

```
<configuration>
    <property>
        <!--指定 SecondaryNamenode 所在地址-->
        <name>dfs.namenode.secondary.http-address</name>
        <value>Master:50090</value>
    </property>
    <property>
        <name>dfs.replication</name>
        <value>3</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/usr/local/bigdata/hadoop-2.7.1/tmp/dfs/name</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:/usr/local/bigdata/hadoop-2.7.1/tmp/dfs/data</value>
    </property>
</configuration>

<configuration>
    <!--指定 HADOOP 所使用的文件系统 schema（URI）-->
    <property>
        <name>fs.defaultFS</name>
        <value>Master:9000</value>
    </property>

    <!--指定 hadoop 运行时产生文件的存储目录-->
    <property>
        <name>hadoop.tmp.dir</name>
        <value>file:/uer/local/bigdata/hadoop-2.7.1/tmp</value>
        <description>Abase for other temporary directories.</description>
    </property>
</configuration>
```

图 4.4: HDFS 配置关键代码

4.2　数据管理实现

数据管理模块负责管理系统扩增出来的司法文本数据。数据管理在前端的具体展现形式就是扩增文件列表，本系统的扩增文件列表界面如图 4.5所示。

图 4.5: 扩增文件列表界面

文件列表展示了用户未删除的扩增数据，每一行就是一份司法裁判文书。对于每一篇文书，UI 都会显示其文件名、文件大小和扩增时间。系统会在用户点击进入数据管理界面的时候到 MySQL 中检索用户扩增过的文本文件信息，找到全部未删除的文件后，将其展示在扩增文件列表中。

用户可以在数据管理模块对自己扩增的文件进行下载和删除操作。文件下载为用户提供了便捷的下载接口，用户仅需点击想要下载文件后的“下载”按钮，就可以将文件下载到本地浏览器。如果文件还未扩增完成，用户可以等待至文件完成扩增后再次进入数据管理界面进行文件的下载。文件删除为用户提供了删除无用文件的接口，用户可以在不想要的文件行内点击“删除”按钮，当删除命令传到后端，系统会到 MySQL 中查询文件的系统文件名和存储位置，然后到 HDFS 中删除文件，并清除数据库中的文件信息。

4.3　文书解析实现

文书解析是为了把用户上传的司法文件转换成数据扩增模块可直接使用的标准 json 格式文件，这一部分也可以看作是文本预处理的过程。文书解析分为格式转换、中文分词和标签提取三个步骤，经过三个步骤的处理，非结构化或者半结构化的数据会被解析成结构化的 json 格式，并被存储到 HDFS 中。文件存储前会根据文书内容计算文件的系统文件名，保证系统文件名不重复，这样可以节省相同文书的重复计算时间，也可以节约文件系统的存储空间。文书解析的关键代码如图 4.6所示。

标准 json 文件的结构化内容有文书号、原文书、文书内容标签和分词结果，具体如表 4.1所示。文书号就是该文书的处理序号，原文书是用户上传的司法文

```
class FormatConversion(FileAnalysis):
    def __init__(self, input_file):
        self.inputfile = input_file
        self.line_dict = {}
    ...
    def format(self):
        sentence_list = super().readfile(self.inputfile)
        for index, line in enumerate(sentence_list):
            self.text_format_byline(line, index)
        super().writefile(self.line_dict)
    #文本格式转换，添加文本序号，sentence 标签与 label 标签
    def text_format_byline(self, line, index):
        line_list = line.strip().split("\t")
        self.line_dict['index'] = index
        self.line_dict['sentence'] = line_list[0]
        self.line_dict['label']= line_list[1]
    ...
```

图 4.6: 文书解析关键代码

本，文书内容标签是时间、地点等可提取的数据标签，分词结果是将句子分割后的词语集。本系统使用 jieba 分词器的精确模式对原文书进行中文分词。

表 4.1: 标准 json 文件的字段格式

字段名	类型	解释
id	int	存储该文书的顺序号
sentence	string	存储该文书的原始内容
label	dict	存储系统提取出的标签以及对应的文本数据
seg_sentence_list	list	存储原始文本分词后的结果，将词语保存在 list 中

4.4　数据扩增实现

数据扩增模块是本系统的核心模块，主要负责把用户上传的原司法文本经过数据处理和加工，扩增成新的高质量司法文本。本模块分为主观质量维度的扩增和客观质量维度的扩增两大部分，具体的系统实现会在本节接下来的内容里介绍。

4.4.1 主观质量维度的扩增实现

司法文本数据质量的度量可以分为主观和客观两个维度。客观质量维度描述了数据在文本构成上的质量好坏，例如信息是否完整、格式是否正确等；主观质量维度则更关注文本在语义质量上的表现，例如用词是否合理，情感倾向是否中立等。客观质量维度和主观质量维度共同构成了司法裁判文书的完整质量维度，对于司法文本，必须满足上述两类质量要求，才可以确保其具备较高的质量。

本系统主观质量维度的扩增方法共有两种，分别是司法语言特征扩增和基于文本情感分析的扩增。

司法语言特征扩增的具体实现为：先从中国裁判文书网上获取一批高质量的裁判文书，对这批文书进行分词操作，然后用机器学习方法 word2vec 学习它们的词向量，每个词的词向量代表了这个词在这批语料中的特征，词向量模型就是这批裁判文书的语言特征模型。训练词向量得到语言模型的关键代码如图 4.7所示。得到 CBOW 词向量模型后，把用户上传的原司法文书输入模型，得到的输出就是根据司法语言特征方法扩增出的数据。

```
sentences=word2vec.Text8Corpus("F:/data/corpuscopy.txt")
model=word2vec.Word2Vec(sentences,size=100, window=5, min_count=1)
model.save('word2vec.model')
model.wv.save_word2vec_format('word2vec.txt',binary = False)
```

图 4.7: 训练词向量得到语言特征模型关键代码

基于文本情感分析的扩增具体实现为：先对训练数据进行标记，对于有明显情感偏向的文本标记 1，无情感偏向的文本标记 0，然后用机器学习的方法训练这批数据。具体训练过程为，先分词，然后用 CountVectorizer 函数将词语和训练数据集向量化，接下来用贝叶斯函数建立分类模型，最后迭代计算得到最终分类模型。本系统把该模型作为分类器来对扩增替换词进行分类，如果某个词被判定为带有主观情感，则将其设为禁用词。训练上述司法文本情感分类模型的关键代码如图 4.8所示。

除了上述通过机器学习模型判断扩增替换词的情感偏向性方法外，本系统还会根据已有的情感词典来辅助判断词语的情感偏向性，对于词典中的非情感中立词，系统也会将其设置为禁用词。这就保证了数据扩增的输出文本在情感偏向性上一定是中立的。

```
#导入 CountVectorizer 函数，将中文词语向量化
from sklearn.feature_extraction.text import CountVectorizer
…
默认参数向量化
vect = CountVectorizer()
term_matrix = DataFrame( vect.fit_transform(X_train.cuted_judicialtext ).toarray(), columns =
vect.get_feature_names() )
…
#导入朴素贝叶斯函数，训练分类模型
from sklearn.naive_bayes import MultinomialNB
nb = MultinomialNB()
# 引入 pipeline 功能，循环运行以上函数
from sklearn.pipeline import make_pipeline
pipe = make_pipeline(vect, nb)
```

图 4.8: 司法文本情感分析分类模型关键代码

4.4.2　客观质量维度的扩增实现

客观质量维度的扩增在主观质量维度的扩增之后，客观维度的扩增对司法文本先后进行延迟性扩增，真实性扩增，完整性扩增，一致性扩增和易读性扩增。每个扩增维度的具体描述如表 4.2所示。

表 4.2: 客观质量维度的扩增描述

扩增方法	**方法说明**	**详细描述**
延迟性扩增	保证事件发生时间和反应时间之间的时间差合理	案件信息延迟性扩增 裁判文书延迟性扩增
真实性扩增	保证文书中司法用语真实	地区真实性扩增 案由真实性扩增 引用真实性扩增
完整性扩增	保证文书结构的完整	司法主体完整性扩增 诉费承担完整性扩增
一致性扩增	保证文本格式是统一的	数字一致性扩增 标点一致性扩增
易读性扩增	保证扩增文本是容易被理解的	句长易读性扩增

延迟性扩增会对待扩增文本的时间标签相关数据进行扩增。案件信息延迟性扩增提取案件发生时间标签和立案时间标签，生成两个时间差较为合理的时间点，确保扩增的案件在案情发生后不久被公安机关立案。裁判文书延迟性扩增提取立案时间标签和审判时间标签，同样生成两个时间差合理的时间点，作

为扩增文本的立案和审判时间。

```
def regionauthenticity_aug(self):
    words = pseg.cut(fact)
    new_fact = ""
    cflag = o
    for word,flag in words:
    ...
    if flag in ['ns', 'nr']:
        #若该地名为县或区，从县区级单位开始选择
        if "县" in word or "区" in word:
            sql = 'select area.name from area join city ON area.cityCode = city.code'\
                'join province ON area.provinceCode = province.code \
                where city.name="%s";'% city_name
            self.cursor.execute(sql)
            word = random.choice(self.cursor.fetchall())
            area_name = word
        #若该地名为市，从市级单位开始选择
        elif "市" in word:
            if province_name:
                sql = 'select city.name from city'\
                    'join province ON city.provinceCode = province.code\
                    where province.name="%s";'% province_name
                self.cursor.execute(sql)
                word = random.choice(self.cursor.fetchall())
            else:
                word = self.city[random.randint(O, self.province.__len__()- 1)]
            city_name = word
        #从最高级别地名单位开始随机选择
        elif "省" in word:
            word = self.province[random.randint(O, self.province.__len__()- 1)]
            province_name = word
    new_fact += word
return new fact
```

图 4.9: 地区真实性扩增关键代码

真实性扩增会提取文书的地点、案由和法条标签进行扩增。地区真实性扩增提取地点标签，系统数据库提前内置了中国的省份、城市和区县名称，三者为从属关系，在进行地名替换时，会随机选择省份名，然后到其下属地名列表中再次随机选择，进行地点标签值的替换。案由真实性扩增提取案由标签，根据案由标签内容选择性替换，案由就是犯罪嫌疑人被法院定下的罪名，本系统的案由被限制在最高院颁布的案件案由规定范围内，保证扩增后文书的相关案由都是

真实存在的。引用真实性扩增提取引用标签，引用法条替换方法和案由替换方法完全一样，所替换的法条也被限制在最高院给出了法条列表中。真实性扩增中的地区真实性扩增关键代码如图 4.9所示。客观质量维度的扩增内的几种扩增方法虽然设计维度不同，出发点也不一样，但是每种扩增方法的大致思路相似，所以这里只展示地区真实性扩增的核心代码，其他部分的实现代码不再赘述。

完整性扩增会提取文书的人名和诉费标签进行扩增。司法主体完整性扩增提取人名标签，在数据库中随机选择新的人名进行替换，司法文书中出现的名字就是司法主体，包括诉讼人、被告人等涉案人员，要注意的是，同一个司法主体不可以出现前后名字不一样的情况。诉费承担完整性扩增提取诉费标签，对诉费承担这个可能存在的部分进行扩增，诉费就是诉讼费用，是指当事人进行司法诉讼时需要向人民法院上交的费用，这部分扩增结果的标准格式是“案件受理费 ×× 元，由 ×× 负担”。

一致性扩增会提取文书的数字和标点符号。对于数字一致性扩增，判断数字是否为阐述判决结果或者分析过程的分项标号，如果是，则用汉子数字，如果不是，就用阿拉伯数字。对于标点一致性扩增，系统限制扩增后的文本中只可以出现《诉讼文书样式》中规定的标点符号。

易读性扩增会统计上述所有扩增方法词语替换前后的词语长度，要求每次替换后词语长度不可以超过替换前词语长度的两倍。这是为了保证扩增后的文本不会太长，控制文本的易读性。这也是高质量裁判文书的一个重要评判指标。

4.5　实验分析

本节首先介绍了数据质量驱动的司法文本数据扩增实验所用的实验数据，然后基于实验数据进行数据扩增实验操作，得到了扩增文本的具体案例，并评估扩增数据的质量，分析得出数据扩增实验的结论。

4.5.1　实验数据集

实验选用 CAIL2018 数据集，该数据集总共包含 18.8 万篇司法裁判文书，文件格式是 json。在 json 文件中，每一行就是一篇司法文书，每篇文书都以字典（Dictionary）格式存储。字典中有多个键值对，键也可以理解为字段，包括案件记录、案由、引用法条等字段。在实验中，主要对司法文书本身进行扩增，即案件记录字段，将其它字段作为扩增辅助数据。实验数据集的具体信息如表 4.3所示。

表 4.3: 实验数据集相关信息

数据集	CAIL2018
数据集格式	json 的字典格式
司法文书数量	18.8 万篇
文书包含字段	案件记录；案由；引用法条；嫌疑人；刑期；罚金
单篇文书长度	约 100-300 字

4.5.2 实验内容

实验在 CAIL2018 数据集中随机选取司法文书，作为输入数据，用数据质量驱动的司法文本数据扩增方法进行扩增。对于某一篇文书，首先会提取其案件记录字段作为扩增对象，然后获取文书相关标签，对文书进行主观质量维度的扩增和客观质量维度的扩增。扩增文本示例如表 4.4所示。

表 4.4: 司法文本数据扩增示例

原文本	公诉机关指控，2016 年 8 月 18 日 00 时许，被告人刘某回到遵化市遵化镇何家园村其租房处，看见房东女儿何某乙离开后，遂生盗窃之意，进入何某乙居住的房中进行盗窃，被祁某等人发现并抓获。针对指控的事实，公诉机关提供了相应的证据，认为被告人刘某的行为已经构成盗窃罪，应依照《中华人民共和国刑法》第 264 条的规定，提请本院依法判处。
扩增文本	杭州市西湖区人民检察院指控，2018 年 5 月 30 日 16 时许，被告人周某回到杭州市西湖区其租房处，发现房主孩子段某出发后，遂生偷窃之意，窜至段某暂住的房中实施盗窃，被胡某看到并制服。对于指控的情况，公诉机关出示了相关证据，认定被告人周某的行为已经构成盗窃罪，应当根据《中华人民共和国刑法》第 264 条的内容，特提请本院予以定罪处刑。

从表 4.4中可以看到，扩增后的司法文本在语法层面是正确的，在语义层面也是完整的。

高质量的数据扩增要求扩增前后数据表达的信息不变，在司法裁判文书中的体现就是，扩增前后文本的司法语义不变。文本的司法语义可以从司法判决中的案由体现，案由就是罪名，如果扩增后的司法文书的案由与扩增前相同，且对于文书而言，案由是合理的，则认为数据扩增是高质量的。本实验从 CAIL2018 数据集中选取了 100 篇司法文本，对其进行数据扩增操作，对于扩增后的司法文本，我们邀请了南京大学法学院的专业同学进行案由判断，判断结果如表 4.5所示。

从表 4.5中可以看到，实验共选取了盗窃，故意伤害，诈骗，虚开增值税专用发票、用于骗取出口退税、抵扣税款发票，强奸五类案由的文书用于扩增。经

表 4.5: 文书扩增前后案由变化

原文书案由	**原文书数量**	**扩增后案由不变的数量**
盗窃	38	38
故意伤害	25	25
诈骗	19	19
虚开增值税专用发票、用于骗取出口退税、抵扣税款发票	10	10
强奸	8	8

南京大学法学院的专业同学判定，扩增后的司法文本都没有改变原文书的案由，且扩增文本的语言逻辑是合理的，这就证明了本实验的数据扩增结果是高质量的。

为了证明本文提出的数据扩增技术可以生成可用性高的司法文本，故将扩增后的文书作为训练数据，去训练司法领域的机器学习模型，用生成的模型进行罪名预测，根据预测结果判断扩增数据的可用性。

本实验随机选取了 16000 条 CAIL2018 数据集中的司法文本，对这批数据进行数据扩增操作，得到案由和扩增前相同的 16000 条扩增数据。此外，也用 Guo 等人提出的 TauJud 框架 [7] 扩增了 16000 条司法文本。实验选用 16000 条原文书、16000 条扩增文书和 16000 条 TaoJud 扩增文书做了对比实验。

具体操作为：选取罪名预测模型作为机器学习模型，使用的神经网络结构为 TextCNN，分别将上述三组数据作为训练数据，对罪名预测模型进行训练，得到三个内部结构不同的模型。在模型训练过程中，输入数据是案件记录字段，标签是案由（即罪名），为了控制变量，也要保证模型的训练次数要一样。

得到三个罪名预测模型后，选取 1000 条 CAIL2018 数据集中的数据作为测试数据，对模型的预测准确率进行统计对比。三组数据集训练出来的模型预测准确率如表 4.6所示。

可以看到，基于扩增文书训练的罪名预测模型预测准确率达到了 80.5%，超过了基于原文书训练的模型。这里需要指出，原文书 CAIL2018 数据集来自于中国裁判文书网，原数据在质量上本身就有保证，而基于扩增数据训练的模型效果能够超过原文书的训练效果，因此可以得出结论：本文提出的数据扩增技术可以生成可用性高的司法文本。对比 TaoJud 扩增文书训练的模型，可以看到，TaoJud 训练的模型预测准确率为 78.4%，而本系统扩增数据训练的模型预测准确率明显高于 TaoJud，因此可以得出结论：相比于已提出的数据扩增技术，本文设计的数据扩增技术可以扩增出质量更高的司法文本。

表 4.6: 不同数据集训练的模型预测准确率

训练数据集	**模型预测准确率**
原文书	79.9%
TauJud 扩增文书	78.4%
本系统扩增文书	**80.5%**

4.6　系统测试

4.6.1　测试目标

对于本文设计的数据质量驱动的司法文本数据扩增系统，本节针对其功能点，设计了详细的测试用例，通过执行用例来测试系统的完整性和可用性。

4.6.2　测试环境

系统测试选取 Win10 作为操作系统，Chrome 作为浏览器，将服务部署在 Linux 服务器上，同时集群主节点也部署在同一台服务器上。集群节点负责提供 DataNode。测试环境具体信息如表 4.7所示。

表 4.7: 系统测试环境

用户操作系统	Windows10
浏览器	Chrome 浏览器
节点操作系统	Ubuntu 16.04.6
节点内存	8GB
编程语言	Python 3.7.3
数据库	MySQL 5.6

4.6.3　功能测试

本小节会对系统需求分析部分提出的系统功能进行测试用例的设计，并从用户的角度完成系统功能测试。为了覆盖系统的全部功能，本次测试设计了司法文本数据上传测试用例、司法文本数据下载测试用例、司法文书解析测试用例、主观质量维度的扩增测试用例、客观质量维度的扩增测试用例和司法文本文件管理测试用例。

表 4.8是司法文本数据上传测试用例，主要测试系统的文件上传功能是否正常，用户是否可以把预备好的司法文书导入到数据扩增流程中。如果用户上传

错误格式的司法文书文件，系统是否可以识别出，并提示用户重新按照系统要求格式上传司法文书。

表 4.8: 司法文本数据上传测试用例

测试 ID	TC1
测试名称	司法文本数据上传
测试功能	用户上传预备好的司法文书，系统自动检测文书格式，符合格式则存储文件准备解析文书
测试步骤	1. 用户依据系统对上传文件的格式要求预备好司法文书 2. 用户登录进入系统 3. 用户点击文件上传 4. 用户选择文件并确认 5. 系统检查文件格式，若不符合则提示用户重新上传，若符合则将文件存储至 HDFS
预期结果	1. 对于步骤 5，系统检测文件格式，若格式不符合系统要求，则不保存文件，且返回提示，要求用户重新上传司法文书 2. 对于步骤 5，系统检测文件格式，若格式符合系统要求，则保存文件至 HDFS，并返回结果

表 4.9是司法文本数据下载测试用例，主要测试系统的文件下载功能是否正常。在数据扩增完成后，是否可以从系统提供的下载接口获取扩增数据，并保证文本信息能被完整下载。

表 4.9: 司法文本数据下载测试用例

测试 ID	TC2
测试名称	司法文本数据下载
测试功能	系统完成数据扩增后，用户点击文件下载按钮，将扩增文本下载到本地
测试步骤	1. 用户选择要下载的文件，点击下载按钮 2. 系统从 HDFS 获取文件信息，将文件下载到用户浏览器
预期结果	用户选择下载文件后，可以从本地浏览器获取到相应的司法裁判文书

表 4.10是司法文书解析测试用例，主要测试系统是否能够把用户上传的原文件解析为 json 格式的标准文件，以及是否能对司法文本进行中文分词和内容标签的提取，对分词和标签提取结果，能否将其正确存入 json 文件。解析得到 json 文件后，是否可以存入 HDFS 供下一步使用。

表 4.11是主观质量维度的扩增测试用例，主要测试系统能否通过 CBOW 模型扩增出具备司法语言特征的司法数据，以及能否通过情感词库扩增出情感中立的司法语句。在完成主观维度的扩增后，是否可以顺利将数据传入客观质量维度的扩增模块。

表 4.10: 司法文书解析测试用例

测试 ID	TC3
测试名称	司法文书解析
测试功能	解析用户上传的司法文书，将其构造成指定格式的 json 文件
测试步骤	1. 系统保存用户上传的原文件后，自动从 HDFS 中读取原文件 2. 将用户上传文件转换成字典格式的 json 文件 3. 对司法文本进行分词，并写入 json 文件 4. 系统自动提取文书内容标签，并写入 json 文件 5. 将解析过程得到的 json 文件存入 HDFS
预期结果	经过文书解析，用户上传的原文件以 json 格式存储在 HDFS 中，待数据扩增模块使用

表 4.11: 主观质量维度的扩增测试用例

测试 ID	TC4
测试名称	主观质量维度的扩增
测试功能	系统对司法文书进行司法语言特征扩增和基于文本情感分析的扩增
测试步骤	1. 系统从 HDFS 中取出解析好的司法文本 2. 系统对文书进行司法语言特征扩增 3. 系统对文书进行基于文本情感分析的扩增
预期结果	1. 对于步骤 2，文书通过 CBOW 模型进行了扩增操作 2. 对于步骤 3，文书经过情感分析保证了文本情感中立性

表 4.12是客观质量维度的扩增测试用例，主要测试系统能否对文书进行延迟性扩增、真实性扩增、完整性扩增、一致性扩增和易读性扩增，以及能否把扩增完成后的数据存入 HDFS。

表 4.13是司法文本文件管理测试用例，主要测试文件管理界面能否完整展示用户扩增过的所有文件的信息，以及用户能否顺利删除不想要的文件。

上述所有测试用例的执行结果如表 4.14所示。测试过程中，严格按照测试用例中的测试步骤一步一步执行，并与预期结果进行对比。可以看到，本节设计的所有测试用例都顺利通过，这证明了系统能够满足需求分析中提出的所有功能点。

4.6.4　性能测试

数据扩增模块是数据质量驱动的司法文本数据扩增系统的核心模块，系统的主要功能就是基于用户输入文本扩增出高质量的司法裁判文书，所以系统的性能测试主要专注于数据扩增模块。

实验对数据扩增模块下的主观质量维度的扩增和客观质量维度的扩增两类

表 4.12: 客观质量维度的扩增测试用例

测试 ID	TC5
测试名称	客观质量维度的扩增
测试功能	系统对司法文书进行延迟性扩增、真实性扩增、完整性扩增、一致性扩增和易读性扩增
测试步骤	1. 从主观质量维度的扩增模块获取司法文书 2. 对文书进行延迟性扩增 3. 对文书进行真实性扩增 4. 对文书进行完整性扩增 5. 对文书进行一致性扩增 6. 对文书进行易读性扩增
预期结果	1. 对于步骤 2，文书完成延迟性扩增 2. 对于步骤 3，文书完成真实性扩增 3. 对于步骤 4，文书完成完整性扩增 4. 对于步骤 5，文书完成一致性扩增 5. 对于步骤 6，文书完成易读性扩增

表 4.13: 司法文本文件管理测试用例

测试 ID	TC6
测试名称	司法文本文件管理
测试功能	用户完成扩增后，文件管理界面显示文件列表，且提供文件删除按钮
测试步骤	1. 用户进入文件管理界面 2. 用户可以在文件列表浏览文件信息，文件信息包括了用户文件名、文件扩增时间和文件大小等 3. 用户点击删除按钮，删除指定文件
预期结果	1. 对于步骤 2，用户能够浏览所有自己扩增的数据文件 2. 对于步骤 3，系统会删除指定文件的所有信息，在文件列表中无法再看见该文件

表 4.14: 测试用例执行结果

测试 ID	**用例描述 ID**	**测试结果**
TC1	UC1	通过
TC2	UC2	通过
TC3	UC3	通过
TC4	UC4	通过
TC5	UC5	通过
TC6	UC6	通过

扩增方法进行了性能测试，两类扩增方法耗时的总和就是系统完整数据扩增方法的总耗时。性能测试结果如表 4.15所示。

表 4.15: 数据扩增性能测试

扩增文本数	耗时/s
100	16.31
500	89.78
2000	392.65

4.7 本章小结

本章详细介绍了数据质量驱动的司法文本数据扩增系统的技术实现。首先是数据交互模块，介绍了文件上传、文件下载和数据展示等功能的技术实现。然后是数据管理模块，描述了文件列表和文件删除的实现方法。接下来介绍了文书解析模块的具体实现，分为格式转换、中文分词和标签提取三个部分。最后是数据扩增的实现，从主观质量维度的扩增和客观质量维度的扩增两个大方向进行技术讲解。对于上述实现的技术，本章基于 CAIL2018 数据集对其进行了实验分析。对于基于数据扩增技术开发的系统，本章也对其进行了系统功能测试和性能测试。

第五章　总结与展望

5.1 总结

我国对于智慧司法的重点建设催生了一系列司法领域的人工智能研究成果。已经有一些基于机器学习的司法推荐系统被设计出来，但是因为各地司法文本的存储方式和质量标准不统一，导致相关机器学习模型使用的训练数据在数量和质量上都有所限制，所以这些推荐系统的应用效果还不够理想。本文设计并实现了数据质量驱动的司法文本扩增系统以应对上述问题。

本文设计的数据扩增技术以及开发的系统是针对司法领域的，扩增过程将传统数据扩增方法和司法领域特征相结合，注重数据扩增的质量，将整个扩增技术分为了主观质量维度的扩增和客观质量维度的扩增。

系统提供的主观质量维度的扩增聚焦司法文本的语义质量，保证扩增文本具备司法语言特征和情感中立性。对于司法语言特征扩增，采用机器学习的方法，学习高质量裁判文书的文字语言特征，并基于训练出的 CBOW 模型进行扩增操作。对于基于文本情感分析的扩增，采用现有的情感词典，并加以学习部分标记情感倾向的司法文书，融合得到一个司法领域的情感词库，在扩增过程中确保扩增词汇在情感词库中的情感标签是中立的，这样就可以保证扩增出的司法文本也是情感中立的。经过试验证明，本系统主观质量维度的扩增技术能够扩增出在语言特征和情感倾向性这两个主观质量维度上，都具备较高质量的司法数据。

系统提供的客观质量维度的扩增针对文书中可结构化提取的部分进行扩增。本文基于客观信息论和司法文书的质量评估研究，将司法文本的客观质量指标分为若干维度，将其中可转换为扩增方法的维度进行解析，确定了数据的延迟性扩增、真实性扩增、完整性扩增、一致性扩增和易读性扩增五种客观质量扩增方法。五种方法分别用于保证扩增数据在司法层面具备案件时间线合理性、案件描述真实性、裁判文书结构完整性、文书前后用语一致性和文书易理解性。

为了让用户能够简单快捷地实现司法文本扩增，本文设计了操作简便的 web 应用，用户可以通过 web 应用对系统进行控制。web 应用采用了 Django 框架，设计成 B/S 架构，上述所有扩增技术都集成在了该应用中。用户可以上传待扩增司法文本到系统中，系统执行完扩增任务后会把扩增信息展示到前端，用户可以选择下载或者删除扩增后的文件。

5.2 未来工作的展望

本文设计并实现了一个数据质量驱动的司法文本扩增系统，该系统的详细设计和技术开发还有可以改进的地方，在后续工作中，可以从以下几个方面进行提升：

第一，基于文本情感分析的扩增方法在情感倾向性判断准确率上还有提升空间。本文进行情感倾向性判断时，使用的词库是由两种方法融合而成的，其中一种是直接使用开源情感词典，另一种是训练少量的标注数据得到分类器。这两种方法的融合虽然在一定程度上能够体现司法领域特征，但是由于训练数据量少，且开源情感词典是面向全领域的，所以在司法领域的应用效果还不够好。如果要提高情感词库在司法领域的应用效果，则需要更多司法从业人员对司法文本的情感倾向性进行标注，在获得大量标注数据后，再用机器学习算法训练数据，才能够生成准确率更高的情感倾向性分类器。

第二，系统支持的输入数据类型还有待扩展。系统当前只支持 json 和文本格式数据的输入，虽然这两种格式的司法文书在司法领域占据了较大的比例，但由于各司法单位数据存储格式可能不同，在实际调研中，也遇到了 XML 格式的司法文本。所以随着有扩增需求的裁判文书类型增多，系统支持的输入数据格式也应该更多。

第三，系统的用户定制化扩增能力还有待提升。当前系统执行数据扩增是对输入文书进行全流程的扩增，即从主观质量维度的扩增到客观质量维度的扩增每一步都执行过去，但是不排除用户希望只执行扩增步骤中的某一步，生成特定样式的司法文书，对于这种需求，未来可以在系统中内置扩增可选择参数，让用户自行选择参数，执行指定方法的数据扩增。此外，有些机器学习模型可能还需要噪声数据（即坏数据），但是本系统只支持高质量数据的扩增，未来可以增加噪声数据的扩增方法，让系统支持噪声数据的定制化扩增。

参考文献

[1] 李飞, 人工智能与司法的裁判及解释, 法律科学: 西北政法大学学报 036 (005) (2018) 32–41.

[2] T. He, H. Lian, Z. Qin, Z. Zou, B. Luo, Word embedding based document similarity for the inferring of penalty, in: International Conference on Web Information Systems and Applications, Springer, 2018, pp. 240–251.

[3] G. Yan, Y. Li, S. Shen, S. Zhang, J. Liu, Law article prediction based on deep learning, in: 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), IEEE, 2019, pp. 281–284.

[4] R. Pham, Y. Stoliar, K. Schneider, Automatically recommending test code examples to inexperienced developers, in: Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, 2015, pp. 890–893.

[5] W. Janjic, C. Atkinson, Utilizing software reuse experience for automated test recommendation, in: 2013 8th International Workshop on Automation of Software Test (AST), IEEE, 2013, pp. 100–106.

[6] X. Lu, B. Zheng, A. Velivelli, C. Zhai, Enhancing text categorization with semantic-enriched representation and training data augmentation, Journal of the American Medical Informatics Association 13 (5) (2006) 526–535.

[7] Z. Guo, J. Liu, T. He, Z. Li, P. Zhangzhu, Taujud: test augmentation of machine learning in judicial documents, in: Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis, 2020, pp. 549–552.

[8] 张舒, 司法文本数据自动化生成系统的设计与实现, Master's thesis, 南京大学 (2020).

[9] 胡云腾, 论裁判文书的说理, 法律适用 3 (2009) 48–52.

[10] 张慧清, 关俭科, 从两大法系判决释理谈我国裁判文书改革, 佛山科學技術學院學報 (社會科學版) 24 (5) (2006) 77–80.

[11] 韩玉刚, 裁判文书说理问题研究, Ph.D. thesis, 山西大学 (2013).

[12] 张彩红, 裁判文书说理的评价指标体系, Master's thesis, 山东大学 (2016).

[13] 杨贝, 裁判文书说理的量化评价——以 2017 年北京市判决书论证质量调查为例, 中国应用法学 (2) (2018) 17–29.

[14] 李嘉伟, 面向法院裁判文书的质量检测技术研究, Master's thesis, 哈尔滨工业大学 (2018).

[15] 廉昊, 面向裁判文书的大数据质量评估平台的设计和实现, Master's thesis, 南京大学 (2019).

[16] 黎宇, 面向民事裁判文书质量评估平台的设计和实现, Master's thesis, 南京大学 (2020).

[17] 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.

[18] Y. Tian, Y. Song, X. Ao, F. Xia, X. Quan, T. Zhang, Y. Wang, Joint chinese word segmentation and part-of-speech tagging via two-way attentions of auto-analyzed knowledge, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8286–8296.

[19] 张桂平, 刘东生, 尹宝生, 徐立军, 苗雪雷, 面向专利文献的中文分词技术的研究, 中文信息学报 24 (003) (2010) 112–116.

[20] 于江德, 王希杰, 樊孝忠, 基于最大熵模型的词位标注汉语分词, Ph.D. thesis (2011).

[21] 单煜翔, 陈谐, 史永哲, 刘加, 基于扩展 n 元文法模型的快速语言模型预测算法, 自动化学报 38 (10) (2012) 1618–1626.

[22] A. Voina, Statistical estimation in hierarchical hidden markov model, Cybernetics and Systems Analysis 50 (6) (2014) 898–912.

[23] J. G. Rogers, H. I. Christensen, A conditional random field model for place and object classification, in: 2012 IEEE International Conference on Robotics and Automation, IEEE, 2012, pp. 1766–1772.

[24] Y. Zhang, Support vector machine classification algorithm and its application, in: International Conference on Information Computing and Applications, Springer, 2012, pp. 179–186.

[25] M. Al-Amin, M. S. Islam, S. D. Uzzal, Sentiment analysis of bengali comments with word2vec and sentiment information of words, in: 2017 international conference on electrical, computer and communication engineering (ECCE), IEEE, 2017, pp. 186–190.

[26] X. Rong, word2vec parameter learning explained, Computer Science.

[27] Z. Xu, T. He, H. Lian, J. Wan, H. Wang, Case facts analysis method based on deep learning, in: International Conference on Web Information Systems and Applications, Springer, 2019, pp. 92–97.

[28] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp. 168–177.

[29] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? sentiment classification using machine learning techniques, arXiv preprint cs/0205070.

[30] P. Melville, W. Gryc, R. D. Lawrence, Sentiment analysis of blogs by combining lexical knowledge with text classification, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009, pp. 1275–1284.

[31] D. Liu, Y. Wang, Y. Ji, H. Tsuchiya, A. Yamashita, H. Asama, Cyclegan-based realistic image dataset generation for forward-looking sonar, Advanced Robotics 35 (3-4) (2021) 242–254.

[32] J. Wei, K. Zou, Eda: Easy data augmentation techniques for boosting performance on text classification tasks, arXiv preprint arXiv:1901.11196.

[33] R. Y. Wang, D. M. Strong, Beyond accuracy: What data quality means to data consumers, Journal of management information systems 12 (4) (1996) 5–33.

[34] D. Aebi, L. Perrochon, Towards improving data quality., in: CiSMOD, Citeseer, 1993, pp. 273–281.

[35] Y. Wand, R. Y. Wang, Anchoring data quality dimensions in ontological foundations, Communications of the ACM 39 (11) (1996) 86–95.

[36] R. Y. Wang, H. B. Kon, S. E. Madnick, Data quality requirements analysis and modeling, in: Proceedings of IEEE 9th International Conference on Data Engineering, IEEE, 1993, pp. 670–677.

[37] M. A. Tate, J. E. Alexander, Web wisdom: How to evaluate and create information quality on the Web, CRC Press, 1999.

[38] X. Zhu, S. Gauch, Incorporating quality metrics in centralized/distributed information retrieval on the world wide web, in: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, 2000, pp. 288–295.

[39] S.-a. Knight, J. Burn, Developing a framework for assessing information quality on the world wide web., Informing Science 8.

[40] J. Cao, X. Diao, T. Wang, et al., Research on some basic problems in data quality control, Microcomputer Information 9 (2010) 12–14.

[41] L. Cai, Y. Zhu, The challenges of data quality and data quality assessment in the big data era, Data science journal 14.

[42] 葛某智, 苏颖, et al., 信息质量研究的现状与未来研究方向, 中国科技资源导刊 1 (5) (2009) 16.

[43] A. Ortiz, Web development with python and django, in: Proceedings of the 43rd ACM technical symposium on Computer Science Education, 2012, pp. 686–686.

[44] T. White, Hadoop-the definitive guide: Storage and analysis at internet scale. 4th (2015).

简历与科研成果

基本情况 许子桓，男，汉族，1997 年 1 月出生，江苏省南通市人。

教育背景

2019.9 ~ 2021.6	南京大学软件学院	硕士
2015.9 ~ 2019.6	南京邮电大学贝尔英才学院	本科

这里是读研期间的成果（实例为受理的专利）

1. **Xu Z**, He T, Lian H, et al. Case facts analysis method based on deep learning[C]// International Conference on Web Information Systems and Applications. Springer, Cham, 2019: 92-97.

致　谢

在毕业论文即将完成之际，我要感谢在研究生阶段给过我帮助和关心的老师、同学、朋友和家人们。

首先要感谢我的导师陈振宇老师和何铁科老师。陈老师是我科研之路的引路人，从最开始初入科研的迷茫，到后来的驾轻就熟，是陈老师教会我科研方法，引导我找到研究方向，并给了我参与各类科研项目的机会，在一个个项目中，我的科研能力得到了提高，探索热情也被激发了出来。在毕业设计的过程中，从选题、开题到写作，陈老师给了我很多建设性的意见，并定期检查论文进度，督促我不断完善毕设。陈老师认真对待科研和生活的态度也给了我很大的启发，他是我学习效仿的标杆，在未来的生活中，我会像陈老师一样找准目标、制定计划，并全力拼搏，做到无愧于心。何老师在我入学后就给了我充足的学习资源，让我能够快速汲取知识，顺利融入科研小组。在实验室项目中，何老师也对我耐心指导，在很多细节处为我答疑解惑，帮助我总结了做项目的方法论。

也要感谢包括陈老师以及何老师在内的南京大学智能软件工程实验室所有的老师和同学。感谢实验室徐剑锋老师在工程项目上对我的指导，教会我如何找准项目需求，以点至面，找到问题的解决方法。感谢实验室廉昊师兄、王栋师兄、黎宇师姐和张舒师姐对我研究思路、写作技巧和实验方案方面的指导。感谢刘佳玮、李卓阳、郭子琛和张朱佩田四位同组同学在毕设选题、写作过程中提供了宝贵的建议，面对科研困难，在他们的帮助下，我总是能顺利攻克。

还要感谢我的室友薛晓波、杨笑和闫超威。尤其是薛晓波，作为同实验室同学兼室友，在我的论文需要法律专业人员协助分析数据时，他找到了南京大学法学院的专业同学，帮助我完成了司法数据的专业性分析。此外，在毕业设计遇到困难，感受到压力时，室友也会主动与我交流，缓解我的压力，我们会制定各自的论文计划，共同前进，一起努力，对于室友的陪伴，我心存感激。

最后要感谢我的家人，我的父母在经济上和精神上给了我极大的支持，让我可以安心地学习。在毕设期间，父母也时刻关心我的论文进展，用无微不至的关爱鼓励我前进。毕设期间外公的身体遇到了比较严重的问题，但外公依然乐观坚强，反而给了我很多受用的建议，我深深地感谢外公对我无私的付出。

最后，感谢参加我毕业论文评审和答辩的专家们，向各位辛勤敬业的学者致敬！