学号：22200107202	成绩：

苏州科技大学

毕业设计（论文）

题　目　劳动仲裁案件的要素抽取与画像构建系统设计与实现

性　质：毕业设计☑　　毕业论文☐

院　　系：	电子与信息工程学院
专　　业：	计算机科学与技术
年　　级：	2022 级
姓　　名：	王雨洁
指导教师：	奚雪峰

二〇二六年

劳动仲裁案件的要素抽取与画像构建系统设计与实现

摘　要

随着经济社会数字化转型的深入，劳动仲裁案件目前呈现数量快速增长与案情日趋复杂的双重特征，传统的人工处理案件的审查模式存在着效率瓶颈、主观经验差异的缺陷。

针对上述问题，本文设计并实现了一个融合规则引擎、Chinese RoBERTa 多任务神经网络与大语言模型的三层混合要素抽取模型，涵盖六大案由类型，共计 70 多个要素字段，在此基础上构建了法律、事实、风险三维度案件画像体系，并基于 FastAPI 与 React 框架开发了集成化的案件辅助分析系统。

实验结果表明，混合策略在案由分类的准确率和要素抽取 F1 值上均优于单一策略基线，三维画像能够直观反映案件核心特征。本文工作为劳动仲裁领域的智能化信息处理提供了可参考的技术方案，后续可在标注数据扩充、NER 头训练增强以及类案裁决预测等方面进一步优化。

关键词　劳动仲裁；要素抽取；案件画像；自然语言处理；多任务学习

Design and Implementation of an Element Extraction and Profile Construction System for Labor Arbitration Cases

Abstract

With the deepening of digital transformation in the economy and society, labor arbitration cases currently exhibit the dual characteristics of rapid growth in case volume and increasing complexity. Traditional manual case review methods suffer from efficiency bottlenecks and variations in subjective experience.

To address these issues, this paper designs and implements a three-layer hybrid element extraction model that integrates a rule engine, a Chinese RoBERTa-based multi-task neural network, and a large language model. The model covers six case types and over 70 element fields. Based on this model, a three-dimensional case profiling system comprising legal, factual, and risk dimensions is constructed, and an integrated case-assisted analysis system is developed using the FastAPI and React frameworks.

Experimental results show that the hybrid strategy outperforms single-strategy baselines in terms of both accuracy in case type classification and F1 score in element extraction. Moreover, the three-dimensional profiling effectively reveals the core characteristics of each case. This work provides a viable technical solution for intelligent information processing in the field of labor arbitration, with future improvements possible through expansion of annotated data, enhanced training of the NER head, and predictive modeling for analogous case adjudication.

Keywords Labor Arbitration; Element Extraction; Case Profile; Natural Language Processing; Multi-task Learning

目 录

第1章 绪论

1.1 研究背景与意义

随着我国经济结构持续转型及劳动关系市场化改革的不断深入，劳动争议案件数量呈现逐年上升趋势。面向“十五五”规划新阶段，规划明确提出要加强就业服务和劳动者权益保障，提升劳动关系矛盾调解仲裁效能[1]。目前不仅案件总量持续增长，案件类型也日趋复杂，从简单的工资报酬争议，逐步发展为工伤保险待遇、未订立书面劳动合同的双倍工资、违法解除劳动合同赔偿金等多元化纠纷类型；部分案件还涉及平台用工、灵活用工等新型用工模式，对劳动保障治理体系提出了更高要求。

作为劳动争议处理的前置程序，劳动仲裁的过程需要承担大量案件的审理工作。然而，当前仲裁实践中面临着突出的效率问题，仲裁员需要从申请书、答辩状、工资单、考勤记录等十多种材料中人工梳理关键信息，对于涉及多名申请人的集体争议案件，仅信息录入与整理工作就可能耗费数小时。调研发现，不少板块反映兼职仲裁员的办案质量不高，案件办理需要专职仲裁员审核，这变相增加了工作负担[2]。因此，将自然语言处理技术引入劳动仲裁领域，实现案件材料的自动化要素抽取与结构化分析，对于提升仲裁工作效率、帮助仲裁员快速把握案件核心特征、促进案件繁简分流具有非常重要的现实意义。

从技术层面来看，近几年以预训练语言模型为代表的深度学习技术，已经在司法领域得到了比较广泛的应用，智能司法也在慢慢从概念走向实际的落地阶段。不过跟刑事和民事诉讼比起来，劳动仲裁这一块的智能化研究还是相对薄弱一些。劳动仲裁案件本身有一些比较独特的文本特点，比如材料的来源多样，表达的方式口语化程度也比较高，而且不同的案由所对应的要素类型也存在结构上的差异。正是这些特点，导致那些通用的法律 NLP 模型不太容易直接拿过来用，因此有必要针对劳动仲裁这个领域去做专门的模型设计和适配工作。

综上所述，本课题选择以劳动仲裁案件作为切入点，主要是研究如何把领域知识融入到要素抽取模型当中，同时构建一个多维度的案件画像体系，并在此基础上开发出一个集成化的辅助分析原型系统。预期这项研究的成果，一方面能够为劳动仲裁工作的标准化和智能化提供一定的技术支撑，另一方面也可以为法律 NLP 在更细分的

垂直领域当中的应用，提供一个可以参考的实践案例。

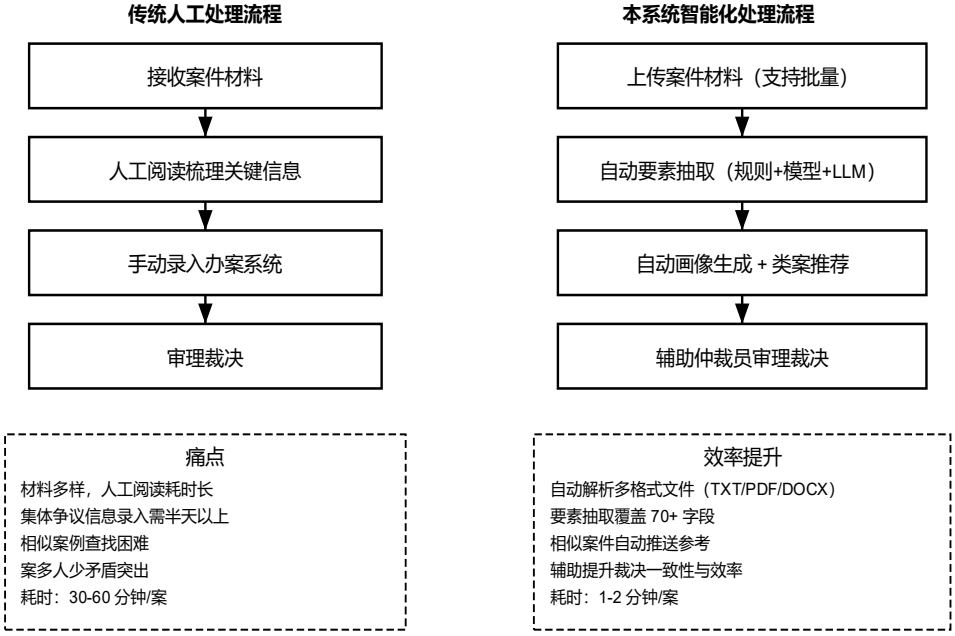

图 1-1 劳动仲裁案件处理流程示意图

1.2 国内外研究现状

本课题的研究涉及法律文本的信息抽取、案件画像构建以及智能司法辅助系统三个方面的技术基础，本节围绕上述方面对相关研究进行梳理。

1.2.1 法律文本命名实体识别与信息抽取

命名实体识别（NER）是法律文本信息抽取的基础任务。早期的方法主要是规则与词典匹配，例如 Dozier 等人[3]针对美国司法判决书构建了基于正则表达式和法律术语词典的实体识别系统，可以识别案件编号、法官姓名、律师事务所等实体。随着机器学习的发展，条件随机场（CRF）模型成为 NER 的主流方法，通过人工设计特征模板实现序列标注。在中文法律 NER 领域，一些研究者[4,5]针对裁判文书的特点设计了融合汉字偏旁部首特征和词边界特征的 CRF 模型，在特定类型实体上取得了优于通用 CRF 的效果。

深度学习的兴起显著推动了法律 NER 的性能提升。双向长短期记忆网络与 CRF 的组合架构（BiLSTM-CRF）避免了人工特征工程的繁琐，通过词嵌入层自动学习文本的分布式表示。在中文法律领域，宿亚杰[6]利用 BiLSTM-CRF 模型对刑事裁判文书

中的作案时间、作案地点、作案工具等实体进行抽取，F1 值较传统 CRF 提升了约 8 个百分点。此后，预训练语言模型（BERT、RoBERTa 等）的引入将 NER 推进到新的阶段。通过在海量通用语料上进行预训练，BERT 能够捕捉丰富的语义和句法信息，仅需在目标领域上进行少量微调即可获得优异的抽取表现。姜中原等人[7]基于 BERT 构建了面向中国刑事裁判文书的实体抽取模型，利用 BERT 的多头注意力机制捕获法律文本中的长距离依赖关系。

在劳动仲裁这一细分领域，信息抽取的研究尚处于起步阶段。与刑事裁判文书不同，劳动仲裁申请书由非法律专业人士撰写的情况更为普遍，文本的规范性较差，实体边界的模糊性更高。同时，劳动仲裁案件涉及当事人信息、工时工资、加班事实、解除原因、社保缴纳等多个要素类别，且不同案由类型对应不同的要素子集，这就要求抽取模型具备对案件类型的自适应能力。已经有研究[8]开始探索基于迁移学习的策略，利用通用法律语料预训练的模型在小样本仲裁数据上进行领域适配，初步验证了该路线的可行性。

1.2.2 多任务学习与联合抽取

传统的信息抽取系统通常将实体识别、关系抽取、文本分类等子任务分别建模，各个模型独立训练。然而，法律文本中各要素之间并非相互独立，例如“违法解除劳动合同”这一案由与“解除原因”“离职时间”“赔偿金计算”等事实要素之间存在强关联。多任务学习（Multi-TaskLearning）通过共享底层表示来捕捉这种关联性，在共享编码器之上设置多个任务特定的输出头，各任务的损失函数加权求和进行联合优化。Collobert 等人[9]最早提出了统一的多任务 NER 架构，后续研究者将其扩展至关系抽取、事件抽取等组合任务。在法律领域，Luo 等人[10]设计了一个多任务模型同时处理法律文本的命名实体识别和法律判决预测，实验表明两个任务通过共享 BERT 编码器实现了相互促进，联合训练后的 NERF1 值和判决预测准确率均优于独立训练。

本课题中涉及的 70 余个要素字段涵盖实体识别（人名、公司名、日期、金额）、文本分类（案由类型、合同类型）、片段抽取（事实描述、加班描述）和数值回归（各项金额）四种不同类型的任务，多任务学习框架天然适用于这一场景。然而，多任务学习中各任务的训练数据量、标注质量和收敛速度可能存在显著差异，如何设置合理的任务权重以及如何处理标注不完整的训练样本，是需要重点解决的技术问题。

1.2.3 预训练语言模型与高效微调

以 BERT、GPT 系列为代表的预训练语言模型已经成为了 NLP 领域的基础设施。在中文法律领域，清华大学基于大量中国裁判文书进行了领域预训练，发布的 Lawformer[11]在多项法律 NLP 任务上显著优于通用领域预训练模型。然而，类似 Lawformer 的大型领域预训练模型参数规模较大（通常达数亿至数十亿），对于计算资源有限的应用场景存在部署困难。

参数高效微调（Parameter-EfficientFine-Tuning）技术的出现为解决这一问题提供了新思路。LoRA[12]通过在预训练模型的权重矩阵旁引入低秩分解矩阵，仅需训练极少量的新增参数（通常为原始参数的 0.1%-1%）即可实现下游任务的适配。Adapter[13]和 PrefixTuning[14]等策略也遵循类似思路。在中文法律 NLP 的实践中，部分研究者[15]探索了使用 LoRA 方法对通用中文大模型进行法律领域的轻量化微调，在下游任务上取得了接近全参数微调的性能，同时大幅降低了训练和部署成本。本课题选用的 QLoRA（QuantizedLoRA）方案在此基础上进一步引入 4-bit 量化技术，使得在消费级 GPU（8GB 以下显存）上对大模型进行微调成为可能。

1.2.4 案件画像与智能辅助系统

案件画像（CaseProfiling）的概念借鉴了用户画像的思想，旨在将案件的各项离散特征整合为结构化的多维描述。在国内司法实践领域，最高人民法院主导推动的“智慧法院”建设已将类案检索与推送作为重点建设内容[16]。上海、浙江等地的法院系统已部署了基于 NLP 的类案推送系统，通过对裁判文书的结构化解构实现相似案件的自动匹配。

在学术研究层面，案件画像的构建方法从早期的基于关键词匹配和规则模板发展到基于深度语义理解的多维特征融合。肖朝霞等人[17]提出了面向民事案件的画像框架，从当事人特征、案件事实、法律适用三个维度构建标签体系。陈宝贵等人[18]将知识图谱技术引入案件画像，通过构建法律概念之间的语义关联图谱来增强画像的表达能力。在可视化方面，仪表盘（Dashboard）和知识图谱可视化已成为展示案件特征的主流形式，让抽象的案件分析结果能够以直观的图形化方式呈现给法律从业人员。

1.3 本文研究内容

本文以劳动仲裁案件为研究对象，围绕“要素抽取—画像构建—系统集成”的核

心链路，开展以下三个方面的研究工作：

⑴ 面向劳动仲裁的多源异构案件材料要素抽取模型设计与实现

针对劳动仲裁案件的申请书、答辩状、证据材料等多源异构文本，研究并构建融合劳动仲裁领域知识的自然语言处理模型。这部分具体工作主要包括以下几点：首先，设计了一个覆盖六大案由类型的要素体系，这六类案由分别是劳动关系纠纷、工伤保险待遇、追索劳动报酬、经济补偿金、赔偿金以及生育保险待遇，整个体系一共包含七十多个要素字段。其次，构建了一个基于正则表达式和关键词匹配的规则抽取器，把它作为整个模型的基础层。然后，又设计了一个基于 ChineseRoBERTa 的多任务联合学习模型作为核心层，这个模型同时处理四类任务，分别是命名实体识别、案由分类、事实片段抽取以及数值回归。此外，还引入了用 Ollama 部署的 Qwen2.5 大语言模型作为增强层，专门用来对那些以自由文本形式出现的仲裁请求事项做结构化的提取。通过这样一套三层混合的架构设计，在保证要素抽取覆盖范围的同时，也提升了模型对语义内容理解的准确性。

⑵ 基于多维要素的劳动仲裁案件画像构建与可视化方法研究

在完成要素抽取的基础之上，本课题进一步研究了如何把那些分散、离散的案件要素进行有机的组织和关联，从而构建出一种能够比较全面且直观地反映出案件特征的结构化“案件画像”。具体来说，课题设计了一套涵盖法律维度、事实维度和风险维度的三维量化评分体系，并在此基础上生成了多层级结构的案件标签树以及关键词云。在可视化方面，还设计了几种不同类型的可视化组件，包括展示关键指标的仪表盘、用于呈现标签分布的标签云、展示要素之间关系的要素关系图，以及用来比较相似案件的柱状图，通过这些组件帮助仲裁员更快地掌握案件的核心特征。

⑶ 集成要素抽取与画像构建的辅助系统实现及应用分析

本课题把要素抽取模型和画像构建的方法做了一个集成，然后基于 FastAPI 后端框架以及 React 前端框架，把整个系统开发出来。这个系统支持批量导入案件材料，并且能够自动完成后续的处理流程，实现从最开始的原始材料、到结构化的要素、再到可视化画像的一站式生成，同时还提供了相似案件匹配的功能。在测试方面，用了真实的历史案件数据来对系统进行验证，评估了要素抽取功能的准确率、召回率和 F1 值，也评估了画像构建方面的完整性和有效性。除此之外，本课题还探讨了这个系统

在一些具体应用场景下面的使用路径和潜在价值，比如案件的繁简分流、相似案例的推荐、以及裁决风险的预警等等。

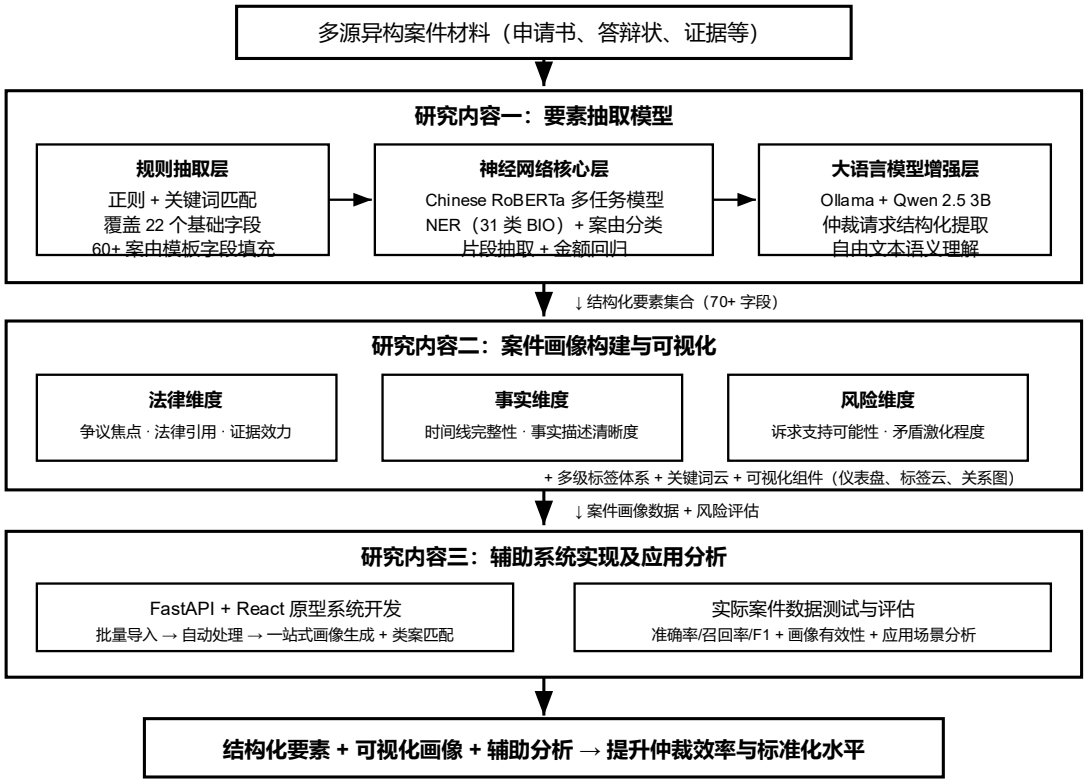

图 1-2 论文整体研究框架图

1.4 论文组织结构

本文共分为六章，各章内容安排如下。

第一章为绪论，阐述劳动仲裁案件智能化处理的研究背景与现实意义，综述国内外相关研究现状，并给出本文的研究内容。

第二章为相关基础知识，介绍系统开发所涉及的技术栈基础、预训练语言模型架构、命名实体识别与序列标注方法、多任务学习框架以及大语言模型本地部署方案。

第三章为需求分析，从目标用户业务流程出发归纳核心需求，进行技术、操作与数据可行性分析，完成业务功能分解与用例建模。

第四章为系统设计，给出系统五大模块的总体架构与接口定义，对各模块进行详细设计，包括三层混合要素抽取架构、三维度画像评分体系、相似案件双策略匹配以及数据库 E-R 模型。

第五章为系统实现，从程序流程、核心代码与界面三个维度阐述各模块的具体实

现，包括多格式文档解析、规则抽取器、神经网络多任务模型、混合抽取器路由策略以及前端组件树。

第六章为系统测试与实验分析，通过消融实验分析各模块的独立贡献，对四种抽取策略进行定性与定量的对比评估，并讨论系统的局限性与改进方向。

最后为结论，总结本文的研究工作与创新点，指出当前不足与后续改进方向。

第2章 相关基础知识

2.1 系统开发框架

2.1.1 FastAPI 后端框架

FastAPI 是一个基于 Python 的现代 Web 框架，它主要就是用来构建那种性能比较好的 RESTful API 的。跟传统的 Flask 还有 Django 这些框架比起来，FastAPI 有三个比较明显的优势。第一个是它本身就支持异步编程这种方式，可以在一个线程里面同时去处理多个 I/O 密集型的请求，这一点对于那些需要等外部服务返回结果的场景特别合适，比如说等 NLP 模型做推理、或者等文件解析完成这些情况。第二个优势是它基于 Pydantic 这个模型，能自动完成请求的校验和序列化，这样一来就可以省掉很多本来要手写的那些样板校验代码。第三个优势是它可以自动生成符合 OpenAPI 规范的交互式文档，也就是 Swagger UI，这个对前后端联调和接口测试来说都挺方便的。本系统选择 FastAPI 作为主要的后端框架，把要素抽取、画像生成、案件管理这些功能都封装成了 RESTful API 的接口，同时还用同样的 FastAPI 框架另外搭建了一个独立的 NLP 模型推理微服务。

2.1.2 React 前端框架与 Recharts 可视化

React 是由 Meta 公司维护的声明式前端 UI 框架，其核心设计理念是通过组件化的方式构建可复用的 UI 单元。React 的虚拟 DOM 机制能够高效地进行差异更新，在处理包含多区域动态渲染的仪表盘页面时具有显著的性能优势。本系统的前端界面基于 React18 搭配 Vite 构建工具开发，采用单页面应用（SPA）架构，通过 AxiosHTTP 客户端与后端进行数据交互。数据可视化方面选用 Recharts 图表库，该库基于 React 组件化思想封装了柱状图、雷达图、散点图等常用图表类型，可直接在 JSX 中声明式调用，与 React 生态高度集成。

2.1.3 MySQL 关系型数据库与 SQLAlchemyORM

MySQL 是应用最广泛的数据库管理系统，其 JSON 数据类型为半结构化数据的存储提供了原生支持。考虑到劳动仲裁案件要素字段随案由类型而变化，JSON 字段的 schema-less 特性使得无需为每种案由建立独立的子表，可以通过 JSON_EXTRACT 等内置函数实现嵌套查询。SQLAlchemy 则是 Python 生态中最成熟的 ORM 框架，通

过对象关系映射将数据库表的行映射为 Python 对象，屏蔽底层 SQL 细节。本系统使用 SQLAlchemy 作为数据库访问层，配合 PyMySQL 驱动连接 MySQL 数据库。

2.2 预训练语言模型

2.2.1 Transformer 与 BERT 基础架构

Transformer 的提出是自然语言处理领域的里程碑事件。其核心的自注意力（Self-Attention）机制通过计算序列中每对位置之间的注意力权重来捕获全局依赖，解决了循环神经网络在处理长序列时的梯度消失和并行计算受限问题。缩放点积注意力（ScaledDot-ProductAttention）的数学形式为：

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \qquad (公式\ 2.1)$$

其中 Q（Query）、K（Key）、V（Value）分别为输入向量经过不同线性变换得到的查询矩阵、键矩阵和值矩阵，d_k 为键向量的维度。

BERT（BidirectionalEncoderRepresentationsfromTransformers）采用 Transformer 的编码器部分作为基础架构，通过在海量无标注文本上进行掩码语言模型（MaskedLanguageModel，MLM）和下一句预测（NextSentencePrediction，NSP）两个预训练任务，学习文本的双向上下文表示。在 MLM 任务中，模型以一定概率随机掩盖输入文本中的部分令牌，并要求模型根据上下文预测被掩盖令牌的原始词汇，从而强制模型学习双向语义依赖。BERT 的预训练-微调范式使得研究者仅需在目标任务的标注数据上进行少量微调，即可获得优异的性能，显著降低了 NLP 任务对大规模标注数据的依赖。

2.2.2 ChineseRoBERTa：面向中文的全词掩码优化

RoBERTa 模型去掉了 NSP 这个训练任务，同时用了更多的训练数据，还采用了动态掩码的策略，所以它在多个 NLP 评测任务上的表现都明显超过了原始的 BERT 模型。在中文的处理场景里面，一个叫做“hfl/chinese-roberta-wwm-ext”的模型又进一步加入了全词掩码的策略，全词掩码也常被写作 WWM。和 BERT 那种按单个汉字进行掩码的方式不一样，全词掩码会把属于同一个词语的所有汉字当作一个整体，要么一起掩码掉，要么就一起保留下来。举个例子，假如有“被申请人违法解除劳动合同”这么一句话，采用 WWM 策略的时候，模型会把“劳动合同”这四个字整体替换

成四个连续的“[MASK]”标记，而不会像之前那样只随机遮盖其中的某几个字。这个策略在处理法律文本的时候尤其重要，因为法律文本里面经常出现大量带有特定法律含义的固定说法和专业术语，比如“违法解除”“劳动合同”“一次性伤残补助金”这些词。通过使用全词掩码，模型可以更好地学习到词语这一级别的完整意思，而不是只学到一些零碎的字面特征。基于这些原因，本课题最终把这个模型选作了要素抽取模块的共享编码器。

2.3 命名实体识别与序列标注

命名实体识别这个任务，经常被简称为 NER，它要做的事情是从没有固定格式的文本里面，找出那些具有特定意思的实体片段，并且给它们分好类。在劳动仲裁案件当中，需要被识别出来的实体类型有不少，比如当事人的姓名、用人单位的具体名称、入职和离职的日期、每个月的工资金额、受理仲裁的机构叫什么名字，还有引用了哪些法律条款等等。对于 NER 来说，一种标准的建模方式就是把它转成一个序列标注问题，具体来说就是给定一串输入的令牌序列 X，可以写成 X=[x_1，x_2，...，x_n]，然后模型要为每一个令牌 x_i 都预测出一个对应的标签 y_i，所有这些标签合在一起就构成了完整的标注方案。

在所有表示实体边界的方法里面，BIO 标注方案是最常用的一种，BIO 是 Begin-Inside-Outside 这几个词的缩写。这种方案会对每一种实体类型都定义两类标签，其中 B-{TYPE}代表这个实体的第一个令牌，I-{TYPE}代表这个实体里面或者结尾处的令牌，而 O 标签则表示当前这个令牌不属于任何实体。本系统使用的 BIO 标签体系一共覆盖了 15 种不同的实体类型，所以总的标签类别数量就是 31 个，计算方法是 15 乘以 2 再加 1。

在 BERT 这类预训练模型上面实现序列标注模型的时候，一种比较典型的做法是这样的：先取出 BERT 最后一层里面每一个令牌对应的隐藏向量，然后把这些向量送到一个线性分类头里面，这个分类头会输出当前令牌属于每一个标签类别的概率分布。训练的时候，一般会用交叉熵作为损失函数，而在解码阶段，虽然可以使用维特比算法来保证整个标签序列在全局上是最优的，但在实际使用 BERT 这种上下文信息很强的编码器时，直接在每个位置上做 argmax 解码通常就能得到比较连贯的结果，所以不一定非得用维特比算法。

2.4 多任务学习框架

多任务学习通常被简称为 MTL，它指的是用一个模型同时去学几个彼此相关的任务，这样做的好处是任务与任务之间的知识可以互相共享，从而帮助每个任务都获得更好的泛化能力。跟单独训练每一个任务的方式比起来，多任务学习的主要优势体现在几个方面：首先，模型中共享的那一层表示结构能够捕捉到不同任务之间的共同特点和联系模式，其次这种方式也能减少对单个任务标注数据量的依赖，此外由于任务之间存在着一种隐式的正则化效果，模型发生过拟合的风险也会有所下降。

在深度学习实践中，多任务学习通常采用“共享编码器+任务特定头”的硬参数共享（HardParameterSharing）架构。其数学形式为：给定输入 X，共享编码器f_θ输出共享表示$h = f_\theta(X)$，各任务 k 通过独立的输出头 g_k 生成预测$\hat{y}_k = g_k(h)$。训练时的总损失函数为各任务损失的加权和：

$$\mathcal{L}_{total} = \sum_{k=1}^{K} \lambda_k \cdot \mathcal{L}_k(\hat{y}_k, y_k), \qquad \text{(公式 2.2)}$$

其中λ_k为任务 k 的损失权重，K 为任务总数。任务权重的设置是影响 MTL 性能的关键超参数，通常依据各任务的训练难度、数据规模和重要程度进行设定。本系统中，NER 任务因标注数据最为稀疏（每案例仅 2-15 个实体片段，绝大多数令牌标签为 O），被赋予最高权重（0.40）；分类任务因其对最终系统决策的关键性次之（0.30）；片段抽取和数值回归任务权重相对较低（各 0.15）。

2.5 参数高效微调与大语言模型推理

2.5.1 LoRA 与 QLoRA

随着预训练语言模型参数规模的增长，全参数微调的计算和存储成本急剧上升。LoRA（Low-RankAdaptation）[19]基于“预训练权重的更新矩阵具有低秩特性”这一假设，通过引入可训练的低秩分解矩阵来实现参数高效的微调。对于预训练权重矩阵$W \epsilon \mathbb{R}^{d\times k}$，LoRA 将更新量$\triangle W_x$分解为低秩矩阵$A \epsilon \mathbb{R}^{d\times r}$和$B \epsilon \mathbb{R}^{r\times k}$的乘积（其中$r \ll \min(d,k)$），冻结原始 W 不变，仅训练 A 和 B。前向传播的计算为：

$$h = W_x + \triangle W_x = W_x + BA_x, \qquad \text{(公式 2.3)}$$

r 的典型取值为 8 或 16，使得可训练参数量仅为原始参数的 0.1%-1%，显著降低

了训练时 GPU 显存的需求。QLoRA[20]在此基础上引入了 4-bitNormalFloat 量化技术，进一步将模型权重的存储从 16/32bit 压缩至 4bit，使得在消费级 GPU（8GB 以下显存）上对参数量达数十亿的大语言模型进行微调成为可能。

2.5.2 Ollama 本地模型部署

Ollama 是一个开源的运行框架，专门用来在本地部署大语言模型，它支持像 Qwen、Llama、Mistral 这些主流开源模型的一键部署和推理功能。Ollama 底层用的是 llama.cpp 这个引擎，因此可以实现比较高效的量化推理，同时它还提供了一个标准化的 HTTP API 接口，方便外部程序去调用它。本系统选择了 Ollama 来部署一个名叫 Qwen2.5:3B 的模型，把它用作大语言模型增强器，由于整个部署都是在本地完成的，所以就不用走云端 API 调用那条路，这样就避免了数据传输带来的延迟问题，也能更好地保护数据的隐私。

第3章 需求分析

3.1 需求概览

劳动仲裁案件辅助分析系统的目标用户群体为各级劳动人事争议仲裁委员会的仲裁员及相关辅助人员。通过对目标用户的业务流程调研与现存痛点分析，本系统的核心需求可归纳为以下四个层面：

⑴ 案件材料的自动化处理需求

仲裁员每天会收到很多不同形式的案件材料，比如手写的或者打印出来的申请书、用人单位提交的答辩状、发工资用的银行流水单、考勤打卡记录的截图、微信聊天的聊天记录、还有工伤认定决定书等。按照现在的处理流程，仲裁员需要把这些材料挨个读一遍，然后手动把里面的关键信息抽出来，再录入到办案系统里面去。如果碰到那种涉及好多个申请人的集体争议案件，光是录入信息这一件事就可能要花掉半天以上的时间。因此，本系统需要具备批量上传多种格式文件的能力，支持的格式包括 PDF、DOCX、TXT 等，还要能自动解析这些文件里面的文本内容，这样才能把仲裁员从那种效率很低的手工录入工作里面解脱出来。

⑵ 案件要素的结构化抽取需求

劳动仲裁案件的处理逻辑高度依赖于若干核心要素的准确认定：当事人身份（劳动者还是用人单位）、入职与离职时间（决定劳动关系存续期间和补偿金计算基数）、工资标准（决定各项待遇的计算基准）、解除原因（决定是经济补偿金还是赔偿金）、以及各项具体的仲裁请求金额。上述要素散落在不同类型材料的各个段落中，人工梳理不仅耗时且容易遗漏。系统需具备从非结构化文本中自动识别并提取上述要素的能力，并按照标准化的要素体系进行结构化组织。

⑶ 案件特征的可视化呈现需求

仲裁员在处理一个新案件的时候，往往需要在很短的时间里面尽快抓住这个案件的一些核心特点，比如这起案件属于哪一种争议类型，相关的重要事实要素是不是都已经有了，申请人提出的诉求在法律上有没有足够的依据，以及这个案件的风险等级大概是什么样子。像这些信息，如果能够用比较直观的图形方式展示出来，比如说做成仪表盘或者标签云那样的形式，就能明显减轻仲裁员在理解和判断时的认知负担，

办案的效率也就能跟着提高。

⑷ 相似案例的检索与参考需求

对于疑难或新型案件，仲裁员通常有参阅历史类似案例的需求，以了解同类型案件的裁决倾向和补偿计算标准。当前的人工查找方式效率低下，系统需基于已抽取的案件要素实现相似案件的自动匹配与推送，为仲裁裁决提供数据参考。

3.2 可行性分析

3.2.1 技术可行性

⑴ 文本解析技术成熟度

PDF、DOCX 等常见文档格式的文本提取已有成熟的 Python 开源库支持：`pypdf`可完成 PDF 文档的文本抽取，`python-docx`可读取 Word 文档的段落与表格内容。上述库经过多年社区维护，在处理中文文档方面具备可靠的稳定性。

⑵ 自然语言处理技术积累

以 BERT 为代表的预训练语言模型已在中文法律 NER 任务上取得了验证性成果。HuggingFace 的 Transformers 开源库为预训练模型的加载、微调和部署提供了标准化的工具链，显著降低了模型工程化落地的技术门槛。本课题采用的 ChineseRoBERTa 模型参数量约 102M，在消费级 GPU（NVIDIARTX3060 及以上）上可完成推理，不存在硬件资源瓶颈。

⑶ Web 开发技术栈成熟度

FastAPI+React 的前后端分离架构已被大量企业级项目验证，开发文档丰富，社区活跃。MySQL 作为持久化存储方案，其 JSON 字段能够灵活应对劳动仲裁案件要素字段的动态变化。

3.2.2 操作可行性

系统设计为 B/S（浏览器/服务器）架构，用户仅需通过浏览器访问前端页面即可使用全部功能，无需安装客户端软件。前端界面以仲裁员熟悉的工作流程为设计参考：上传材料→查看要素→浏览画像→参考类案。交互流程直观，培训成本低。系统同时提供 Swagger 自动生成的 API 文档，便于后期与仲裁机构现有的办案系统进行对接集成。

3.2.3 数据可行性

本课题在做研究和开发的时候，用的是大约 1300 多条经过脱敏处理后的真实劳动仲裁案件材料，这些案件覆盖了几种主要的案由类型，比如追索劳动报酬、工伤保险待遇、经济补偿金、赔偿金，还有劳动关系确认这些。数据的来源主要是已经公开的仲裁裁决文书和一些模拟出来的案例文本，所以不存在泄露个人隐私信息的风险。虽然目前的数据量看上去不算大，原始案例只有 33 条，但我们可以通过一些数据增强的方法来进行扩充，比如替换同义词、做实体掩码的变体，或者用大语言模型来生成一些合成的案例，这样就能把训练集扩大到 150 条以上，从而满足轻量级模型在微调阶段对数据量的基本要求。

3.3 业务功能分解

依据 3.1 节的需求分析，将系统功能分解为以下五个核心功能模块。各模块之间按照“材料→要素→画像→匹配→展示”的数据流方向依次衔接。

⑴ 案件材料管理

文件上传：支持 TXT/PDF/DOCX 格式的单个或批量上传

材料解析：自动检测文件格式并提取文本内容，支持中文编码的自动识别

材料去重：基于 SHA256 指纹检测重复上传的文件

材料预览：在界面中展示解析后的纯文本内容，供用户核验

⑵ 要素自动抽取

规则抽取：基于正则表达式与关键词匹配，批量提取日期、金额、案号等结构化信息

模型抽取：基于 ChineseRoBERTa 多任务模型，自动识别案由类型、合同类型、用工形式等分类要素

LLM 增强抽取：通过 OllamaQwen 模型，对仲裁请求事项等自由文本段进行结构化提取

要素编辑：支持用户对自动抽取结果进行在线修正与保存，修正记录纳入版本历史

⑶ 案件画像生成

三维度量化评分：自动计算法律维度、事实维度和风险维度的评分（0-100）

标签体系构建：基于已抽取要素自动生成多层级案件标签（维度→类型→具体值）

关键词提取：基于词频分析从原始文本中提取案件关键词

风险等级判定：根据风险维度得分自动划分风险等级（低/中/高）

⑷ 相似案件匹配

要素标签匹配：基于 Jaccard 相似度在案例库中检索标签相似的历史案件

向量语义检索：基于 Sentence-BERT 向量在 Qdrant 中检索语义相似案件（可选模块）

三维加权排序：综合法律维度、事实维度和风险维度的相似度进行加权排序

类案详情对比：展示查询案件与匹配案件在要素层面、画像层面和裁决结论层面的差异

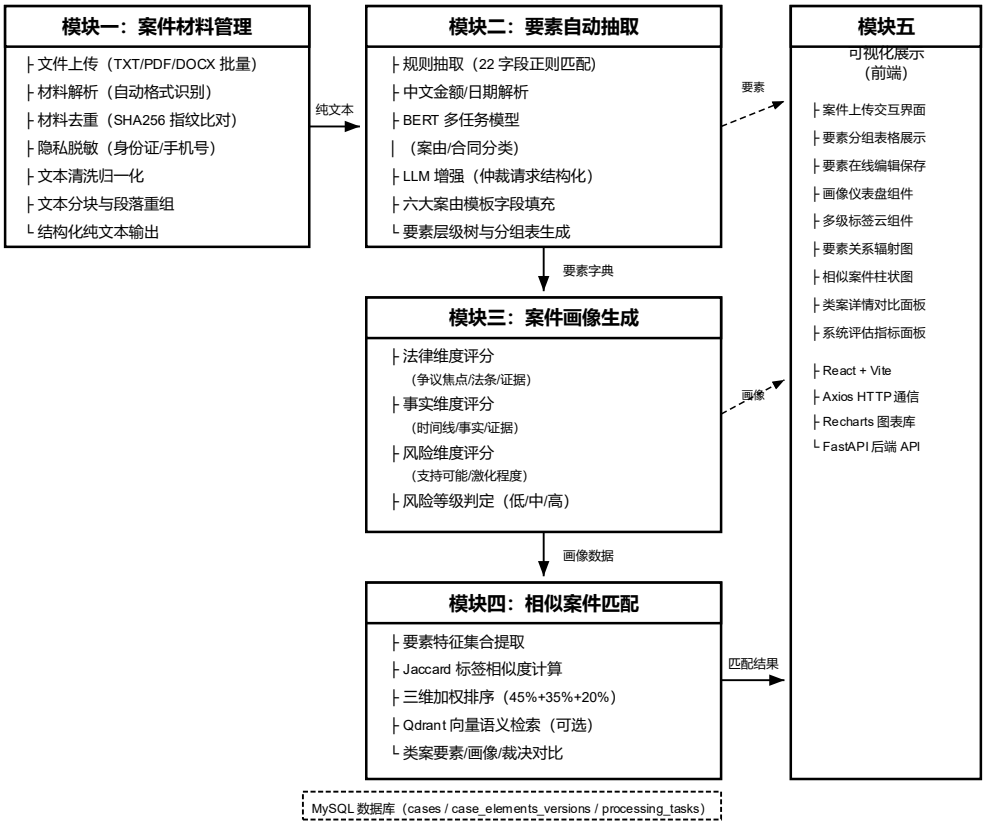

图 3-1 系统功能分解图

⑸ 可视化展示

要素分组表格：按案由类型和要素层级分组展示结构化要素

画像仪表盘：以卡片和进度条形式展示三维度评分

标签云：以字号和颜色编码展示多层级案件标签

要素关系图：以辐射图展示案件核心要素之间的关联

相似案件柱状图：以图表形式展示 Top-K 相似案件及其相似度得分

系统评估面板：展示要素抽取与画像生成的性能指标

3.4 功能性需求分析

本节采用用例驱动的分析方法，识别系统的参与角色及其功能交互，并通过用例图和用例规约对核心功能进行结构化描述。

3.4.1 角色识别

系统涉及两类主要参与者：

仲裁员（主要用户）：劳动人事争议仲裁委员会的工作人员，负责案件的审理与裁决。仲裁员使用本系统完成案件材料的上传、要素抽取结果的查看与修正、案件画像的浏览、以及相似案件参考信息的获取。仲裁员关注的核心指标是：信息抽取的准确性、操作的便捷性、分析结果的可信度。

系统管理员（辅助角色）：负责系统的运行维护，包括模型更新、数据备份、异常任务处理等后台管理操作。管理员的交互频次远低于仲裁员，但涉及的操作权限更高。

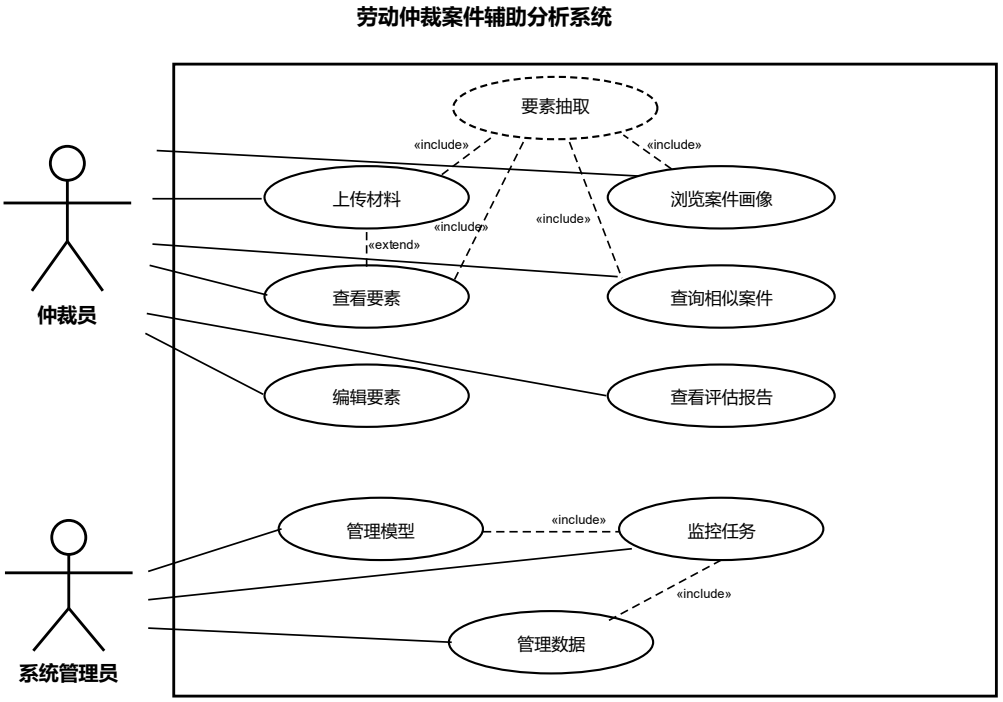

图 3-2 系统用例图

3.4.2 核心用例分析

⑴ 上传并处理案件材料

项目	内容
用例编号	UC-01
用例名称	上传并处理案件材料
参与者	仲裁员
前置条件	系统正常运行，MySQL 数据库可连接，模型服务可访问
后置条件	案件材料存入数据库，结构化要素数据生成，案件画像生成
基本流程	1.仲裁员在界面上输入案件名称，选择待上传文件（可多选） 2.系统校验文件格式，对非支持格式给出提示 3.系统计算文件 SHA256 指纹，与已有记录比对去重 4.系统解析文件内容为纯文本 5.系统调用要素抽取引擎进行规则抽取 6.系统调用 BERT 模型进行案由分类等语义抽取 7.系统调用 Ollama 增强仲裁请求抽取 8.系统将要素字典存入数据库的 elements 字段 9.系统调用画像模块生成三维度评分和标签体系 10.系统返回处理完成状态与案件 ID
异常流程	4a.文件解析失败：记录失败原因并提示用户更换文件格式 5a.规则抽取异常：输出空要素字典，不阻断后续流程 7a.Ollama 服务不可用：跳过 LLM 增强，保留规则与模型结果
补充说明	对于批量上传场景，系统为每个文件创建独立的 CaseFile 记录，所有文件共享一个 Case 主记录

⑵ 浏览案件画像与相似案件

项目	内容
用例编号	UC-02
用例名称	浏览案件画像与相似案件
参与者	仲裁员
前置条件	案件已完成要素抽取（UC-01 执行完毕）
后置条件	无数据变更（只读操作）
基本流程	1.仲裁员在案件列表中点击目标案件 2.系统加载该案件的要素数据，以分组表格展示 3.仲裁员切换至“案件画像”标签页 4.系统展示三维度评分仪表盘（法律/事实/风险）、标签云、关键词列表 5.仲裁员切换至“相似案件”标签页 6.系统计算当前案件与数据库中所有历史案件的 Jaccard 相似度 7.系统按三维加权相似度降序排列，返回 Top-K 案件 8.仲裁员点击某个相似案件，展开详情对比面板 9.系统展示两案件在要素、画像和裁决结论上的差异

异常流程	6a.数据库中无历史案件：展示“暂无相似案件”提示 6b.相似度最高值低于阈值（如<0.1）：标注“匹配度较低”
补充说明	默认 Top-K=5，用户可调整为 3/10/20

⑶ 在线修正要素并保存

项目	内容
用例编号	UC-03
用例名称	在线修正要素数据
参与者	仲裁员
前置条件	案件已完成要素抽取，要素数据已展示在界面中
后置条件	修正后的要素数据存入数据库，生成新版本记录
基本流程	1.仲裁员在要素表格中定位需要修正的字段 2.仲裁员修改字段值（文本编辑/下拉选择/日期选择） 3.系统实时校验字段格式（如日期格式、金额数值） 4.仲裁员点击“保存”按钮 5.系统将修改后的完整要素字典写入数据库 6.系统在 case_elements_versions 表中插入新版本记录 7.系统重新触发画像模块，更新三维度评分和标签
异常流程	3a.日期格式不合法：字段标红并给出格式提示（如“YYYY-MM-DD”） 3b.金额包含非数字字符：提示用户去除单位字符 5a.数据库写入失败：回滚事务，提示用户稍后重试
补充说明	要素版本记录包含版本号（自增）、编辑者标识和编辑时间

3.4.3 非功能性需求

⑴ 性能需求

单个案件（材料合计 10 页以内）从上传到要素抽取完成的响应时间不超过 60 秒（含 NLP 模型推理时间）；批量处理 5 个以上案件时采用异步任务模式，用户无需等待全部完成即可查看已完成案件的结果。前端页面首屏加载时间不超过 3 秒。

⑵ 准确性需求

案由分类的准确率不低于 85%（以人工标注为基准）；核心要素字段（当事人姓名、日期、月工资、仲裁请求金额）的抽取 F1 值不低于 0.80；在规则抽取与 BERT 模型均可用时，混合策略的 Micro-F1 值不应低于任一单独策略。

⑶ 可用性需求

系统提供的 API 接口可用性不低于 99%（非计划性停机），模型服务支持 GPU 不可用时的 CPU 回退推理（响应时间相应延长）。前端界面兼容 Chrome 和 Edge 浏览器的近两个主要版本。

⑷ 可扩展性需求

要素抽取模块的接口设计支持新增抽取策略而无需修改调用方代码；案由类型模板（`case_elements_hierarchy.json`）支持在不修改代码的情况下扩展新的案由分支和要素子字段；系统支持配置化切换抽取模式（纯规则/纯 BERT/纯 LLM/混合）。

第4章　系统设计

4.1　系统模块体系

本系统主要是围绕劳动仲裁案件的一条核心业务链路来设计的，这条链路大致是“材料接入—要素抽取—画像构建—辅助应用”这几个环节，整体上采用了分层模块化的架构方式。系统一共分成了五个一级功能模块，分别是数据接入与预处理模块、要素抽取模块、案件画像构建模块、相似案件匹配模块，还有可视化展示模块。模块和模块之间通过提前定义好的接口来交换数据，功能上彼此之间是解耦的，也就是说一个模块改了不会太影响别的模块，但在处理流程上它们又是互相配合的，共同完成从原始案件材料到结构化案件画像的一站式处理。

划分这几个模块的主要依据，是看每个处理阶段做的事情不一样。数据接入模块负责把那些来源不同、格式也不同的材料，比如 PDF、DOCX、TXT 这些，统一转成结构比较清楚的纯文本。要素抽取引擎模块是整个系统的核心计算部分，它的任务是从那些没有固定格式的文本里面，抽取出结构化的案件要素。案件画像模块会基于已经抽取出来的要素，从多个维度去做量化分析，并且生成一些标签。相似案件匹配模块利用要素的特征，去历史案例里面做检索和对比。最后可视化模块负责把前面分析得到的结果，用比较直观的图形化形式展示给用户看。

上面说的这五个模块之间，其实存在一种“流水线加总控”的双重关系。如果从数据流动的角度来看，案件材料是按照“预处理→抽取→画像→匹配→展示”这个方向单向流动的，也就是说上一个模块的输出结果，会成为下一个模块的输入内容。如果从控制流的角度来看，后端的那个主控程序，也就是基于 FastAPI 的应用，起着调度中枢的作用，它会统一管理每一个模块的调用顺序，还有出现异常情况时该怎么处理，而前端则是通过 RESTful API 和后端的这个主控程序进行请求和响应式的交互。

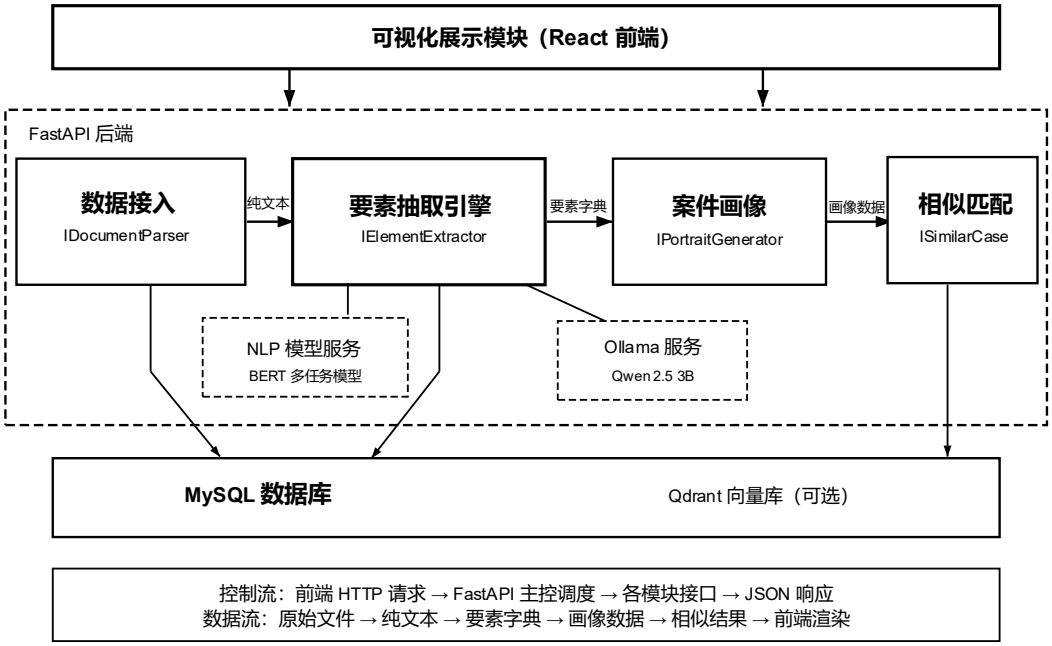

图 4-1 系统总体架构图

模块间的接口设计遵循“声明与实现分离”的原则，每个模块对外暴露一组功能声明（接口），其内部实现可独立演化而不影响其他模块。表 4-1 列出了系统核心接口的定义。

表 4-1 系统核心接口的定义

接口名称	所属模块	主要方法声明	调用方法
IDocumentParser	数据接入	parse(file_path)->str	主控程序
IElementExtractor	要素抽取	extract(text)->dict	主控程序、画像模块
IPortraitGenerator	画像构建	generate(elements,text,evidence_count)->dict	主控程序
ISimilarCaseMatcher	相似匹配	find_similar(case_id,top_k)->list	主控程序
IModelInference	模型推理	predict(text)->dict	要素抽取模块

其中，‘IElementExtractor’作为系统的核心接口，其设计充分考虑了多策略集成的需求。该接口声明了‘extract(text:str)->dict’方法，其返回值为标准化的要素字典，字段集合由‘case_elements_schema.json’定义。当前系统中，该接口有三个具体实现类：‘RuleBasedExtractor’（基于正则规则的抽取器）、‘ChineseLegalElementExtractor’

（基于 ChineseRoBERTa 多任务模型的抽取器）、以及‘HybridExtractor’（融合规则与模型的混合抽取器）。三个实现类遵循同一接口契约，使得主控程序可以在不同模式下通过配置切换抽取策略而不修改业务逻辑代码。

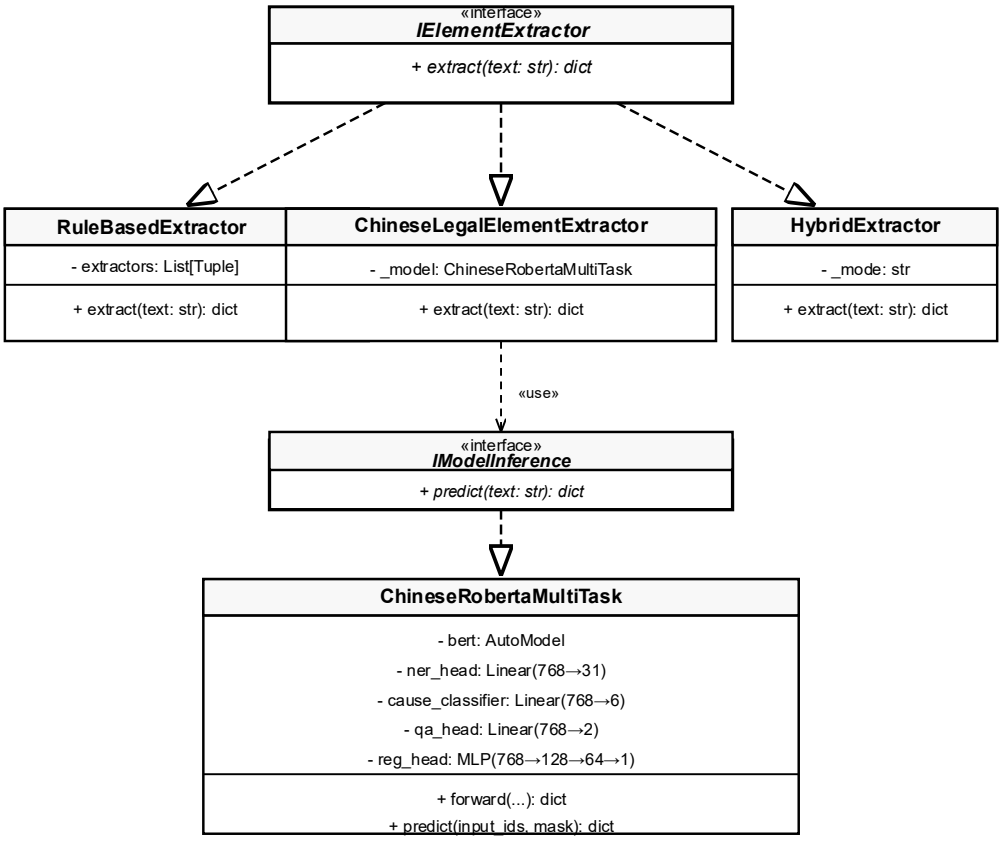

图 4-2 接口与实现类关系图

4.2 数据接入与预处理模块

数据接入与预处理模块做的事情是接收用户上传的案件材料文件，然后完成格式解析、文本清洗，还要做去重和校验，这样就能给后面的要素抽取环节提供一份比较规范、干净的纯文本输入。这个模块一共支持三种常见的文件格式，分别是 TXT 纯文本、PDF 文档和 Word 文档。

这个模块的核心处理流程大致是这样的：用户先通过前端界面上传一个或者多个案件材料文件，后端收到这些文件之后，会先计算每个文件的材料指纹，这个指纹用的是 SHA256 哈希值，然后拿这个指纹去跟数据库里面已经存好的记录做比对，目的是看看有没有重复的文件。对于那些没有重复的文件，系统会根据文件的扩展名来选择对应的解析器去处理，具体来说就是 TXT 文件直接用 UTF-8 编码去读取里面的内

容，PDF 文件则通过一个叫‘pypdf’的库来提取文本，而 Word 文件会用‘python-docx’这个库一段一段地把段落文本提取出来。解析完得到的文本还会经过统一的清洗处理，主要包括把换行符归一化，也就是把‘\r\n’统一换成‘\n’，然后把连续出现的空白行合并成一行，再把开头和结尾的空白字符去掉。最后，清洗干净之后的文本会跟文件本身的元信息放在一起存入数据库，这些元信息包括文件名、文件大小和上传时间，同时这些文本内容也会被送到要素抽取模块那里去做后续的处理。

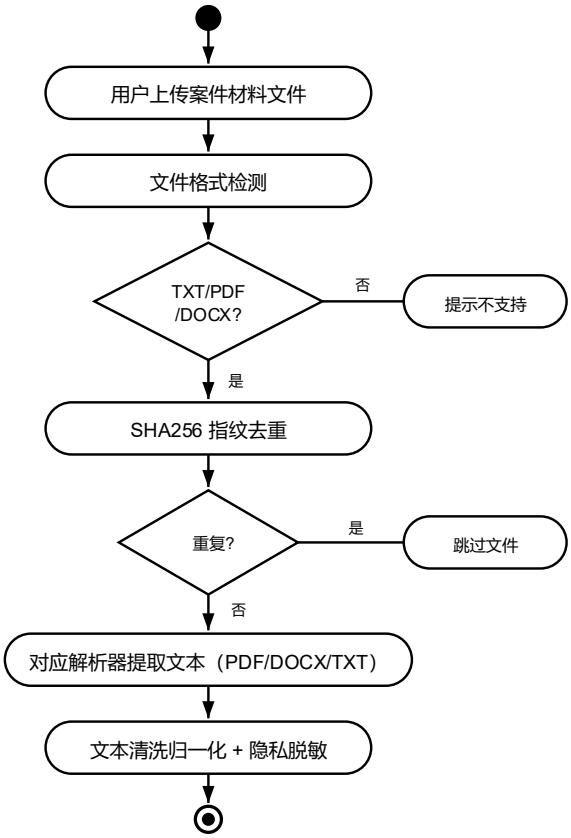

图 4-3 数据接入与预处理模块活动图

这个模块对外暴露了一个叫“IDocumentParser”的接口，里面声明了一个方法，写法是`parse(file_path:str) -> str`。接口设计的关键想法是，把多种文件格式带来的差异都封装在模块内部，对外只提供一个统一的纯文本输出结果。这样一来，如果以后需要支持新的文件格式，只需要新增一个对应的解析器实现就行了，下游的那些模块完全不用改动任何代码。

在文本清洗这个环节里面，系统还特别针对劳动仲裁案件材料的特点，额外增加了一步隐私信息的脱敏处理，具体做法就是用正则表达式去匹配身份证号码、手机号

码以及 Base64 编码的身份哈希值，身份证号码一般是 18 位数字的模式，手机号码是 11 位数字的模式，匹配到之后就用占位符把它们替换掉。需要说明的是，这个处理并不是因为系统功能上非要这么做不可，而是考虑到系统里面存的一些案例数据，以后可能会被拿去做学术研究，比如用于数据集的共享，所以就提前把隐私保护的机制给做进去了。

4.3 要素抽取引擎模块

要素抽取引擎这个模块，是整个系统里最核心的算法部分，它的任务是把那些没有固定格式的仲裁案件文本，转成结构比较清晰的要素集合。设计这个模块的时候，主要目标是既要保证抽取的覆盖面够广，大概有 70 多个要素字段，覆盖了六大类案由，同时还要兼顾抽取的精度和处理效率。为了达到这个目标，模块采用了一种三层混合的架构，具体来说就是“规则基础、深度学习增强、大语言模型补充”这三层结合在一起。

4.3.1 模块总体设计

要素抽取引擎的内部结构一共包含三个子模块，分别是规则抽取器、神经网络抽取器和大型语言模型增强器，其中规则抽取器在代码里叫 RuleBasedExtractor，神经网络抽取器叫 ChineseRobertaMultiTask，大语言模型增强器叫 OllamaClaimsExtractor。规则抽取器起到基础层的作用，它会覆盖全部 70 多个要素字段里面那些具有明确文本模式的结构化信息，这样就能保证基本的抽取覆盖率和输出结果比较确定。神经网络抽取器是核心层，它借助预训练语言模型的语义理解能力，对案由类型、合同类型这些需要结合上下文去做推理的分类任务来进行预测。大语言模型增强器则是一个补充层，主要针对仲裁请求事项和案由模板里的扁平要素当中那些表达方式变化较多、不太好用固定规则去覆盖的自由文本段落，来做进一步的提取。

这三个子模块是通过一个叫 HybridExtractor 的类来统一调度的，调度的策略是按照字段的类型来走不同的路由。具体来说，对于像 case_number、filing_date、arbitration_org 这类结构比较固定的字段，会优先采用规则抽取器的结果；对于分类字段，比如 tmpl_primary_cause、contract_type、employment_type 这些，会优先采用神经网络模型的预测结果；而对于像 claims、gen_facts_and_reasons 这样的自由文本字段，如果启用了 Ollama 就会用大语言模型来提取，如果没有启用的话，就回退到

规则抽取的方式，这个调度逻辑可以参考图 4-4。

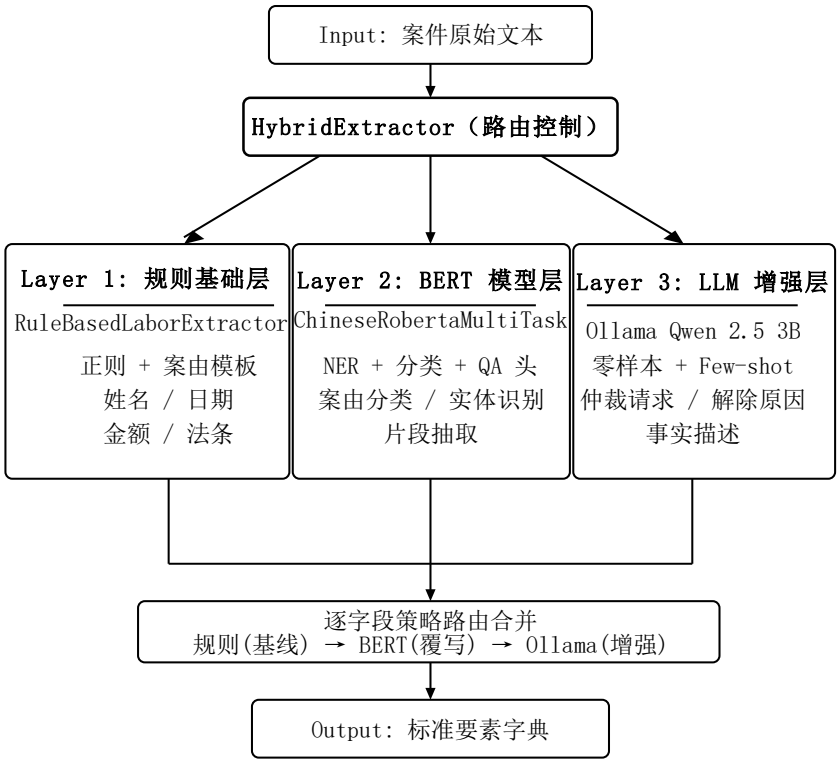

图 4-4 要素抽取引擎内部架构图

4.3.2 规则抽取器设计

规则抽取器主要是靠正则表达式和关键词匹配来实现的，里面一共包含了 22 个基础的抽取函数，以及 60 多个用于模板字段填充的规则。每一个抽取函数都对应一个或者一组在语义上比较相近的要素字段，设计这些函数的时候，主要是先归纳出法律文书中那些比较常见的表达模式，然后针对这些模式设计相应的匹配规则，具体可以参考图 4-5。

拿日期抽取来举个例子，仲裁文书里面日期的写法有很多种，比如“2020 年 3 月 1 日”“2020.03.01”“2020/3/1”“2020-03-01”这些，规则抽取器会统一用一个正则表达式去匹配，这个正则大致是“([12][0-9]{3})\s*[年./-]\s*([01]?[0-9])\s*[月./-]\s*([0-3]?[0-9])\s*(?:日)?”，匹配到之后再用`parse_date_any()`函数统一转成“YYYY-MM-DD”这种格式。对于金额字段，规则抽取器同时支持阿拉伯数字和中文大写数字这两种写法，比如“8000 元”和“八千元”都能解析，其中中文大写数字是通过构建一

个中文数字到单位之间的映射表来实现数值转换的。

案由分类在规则抽取器里面算是一个比较关键的环节，这个模块先定义了八组跟案由相关的关键词，然后根据文本中这些关键词的共现情况，做加权匹配来确定案由。不过基于关键词来做案由分类，局限性也是比较明显的，主要问题在于当一个案件里面同时涉及好几种争议类型的时候，光靠关键词的优先级排序可能就会判断出错。也正是因为这个局限性，后面才考虑引入了深度学习模型。

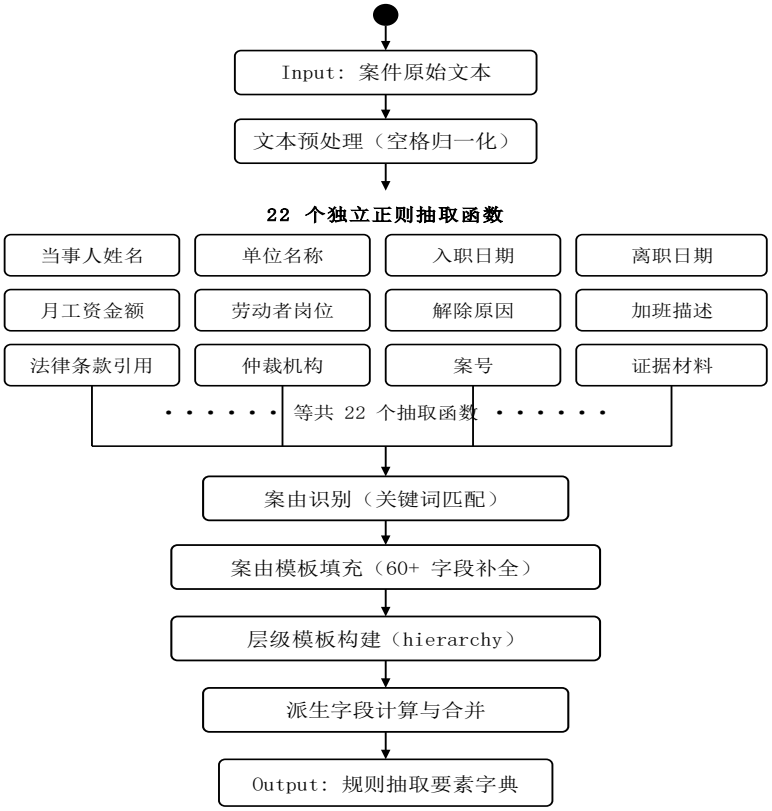

图 4-5 规则抽取器处理流程图

4.3.3 神经网络抽取模型设计

鉴于规则抽取器在语义理解方面的固有不足，系统设计了一个基于预训练语言模型的多任务神经网络，专门针对劳动仲裁领域进行微调。该模型的设计出发点在于：劳动仲裁案件要素的抽取本质上是多种自然语言理解任务的组合——实体识别（当事人姓名、公司名称、日期、金额）、文本分类（案由类型、合同类型、用工形式）、文本片段抽取（事实描述、加班描述、解除原因）以及数值回归（各项金额预测）。与其分别训练多个独立模型，不如利用预训练语言模型的共享表示能力，设计一个多

任务联合学习架构，使各任务之间通过共享编码器相互促进。

模型选用`hfl/chinese-roberta-wwm-ext`作为共享编码器。该模型在中文 BERT 的基础上引入了全词掩码（WholeWordMasking）策略，在预训练阶段以词为单位而非以字为单位进行掩码，从而更好地建模中文词语的完整性语义。对于法律文本中大量出现的专业术语（如“违法解除劳动合同”“一次性伤残补助金”），全词掩码策略有助于模型学习词语级别的语义组合。编码器参数量约为 102M，能够在消费级 GPU（4GB 以上显存）上完成推理。

联合训练的损失函数为各任务损失的加权和：

$$\mathcal{L} = \lambda_{ner} \cdot \mathcal{L}_{NER} + \lambda_{cls} \cdot \mathcal{L}_{CLS} + \lambda_{qa} \cdot \mathcal{L}_{QA} + \lambda_{reg} \cdot \mathcal{L}_{REG}, \quad \text{(公式 4.1)}$$

其中，$\mathcal{L}_{NER}$为令牌级交叉熵损失（忽略标签为-100 的位置），$\mathcal{L}_{CLS}$为各分类头交叉熵损失之和，$\mathcal{L}_{QA}$为起始/结束位置的交叉熵损失平均，$\mathcal{L}_{REG}$为 MSE 损失。任务权重依据各任务在最终系统中的作用和训练难度进行设定：λ_{ner}=0.4，λ_{cls}=0.30，λ_{qa}=0.15，λ_{reg}=0.15。NER 头因标注数据最为稀疏（每案例仅 2-15 个实体片段），被赋予最高权重以加强其训练信号。

4.3.4 模型训练策略

训练用的原始数据来自 33 条真实的劳动仲裁案件材料，这些材料先做了数据清洗和自动标注，之后又用了三种策略来做数据增强。第一种策略是替换法律术语里面的同义词，比如把“工资”换成“薪资”或者“劳动报酬”，这样能增强模型对领域里不同词汇变体的适应能力。第二种策略是做一些实体掩码的变体，具体做法是把人名、公司名、日期这些内容替换成同类的占位符然后再重新生成，目的是增强模型对实体模式的泛化能力。第三种策略是通过 Ollama 部署的 Qwen2.5 3B 模型去生成一些合成案例文本，用来增加训练样本的多样性。经过数据增强之后，训练集大概有 150 条数据，然后按照 7:1.5:1.5 的比例分成了训练集、验证集和测试集。

训练的时候用的是 AdamW 优化器，学习率设成了 2e-5，批量大小是 4，同时用梯度累积把有效批量大小变成了 8，梯度累积的步数设为了 2。学习率调度方面采用了线性预热的方式，预热比例是 10%，并且在验证集的损失上用了早停机制，早停的耐心值设为 3 个 epoch。为了防止在小样本的情况下出现过拟合，设置了权重衰减为

0.01，还用了 dropout，dropout 的概率是 0.1。训练过程使用了混合精度，也就是 FP16，这样可以加快计算速度。

训练完的模型会以`pytorch_model.bin`这种格式保存下来，文件大小大概是 391MB，然后通过`nlp-service`微服务对外提供 HTTP API 的推理接口。

4.3.5 大语言模型增强

对于仲裁请求事项中表达形式高度多变的文本段落——如“请求被申请人支付拖欠的 2023 年 5 月至 6 月工资共计 8000 元，并支付违法解除劳动合同赔偿金 24000 元”——规则抽取与 BERT 模型均存在覆盖不足的问题。系统引入 Ollama 部署的 Qwen2.53B（参数量 30 亿）大语言模型作为补充增强器。增强策略采用结构化提示（StructuredPrompting），在提示中明确要求模型以 JSON 格式输出`{“items”:[...]，“amounts”:[...]}`，并通过`temperature=0`确保确定性输出。当 LLM 输出无法正确解析为 JSON 时，系统自动回退至规则抽取结果，确保服务的鲁棒性。

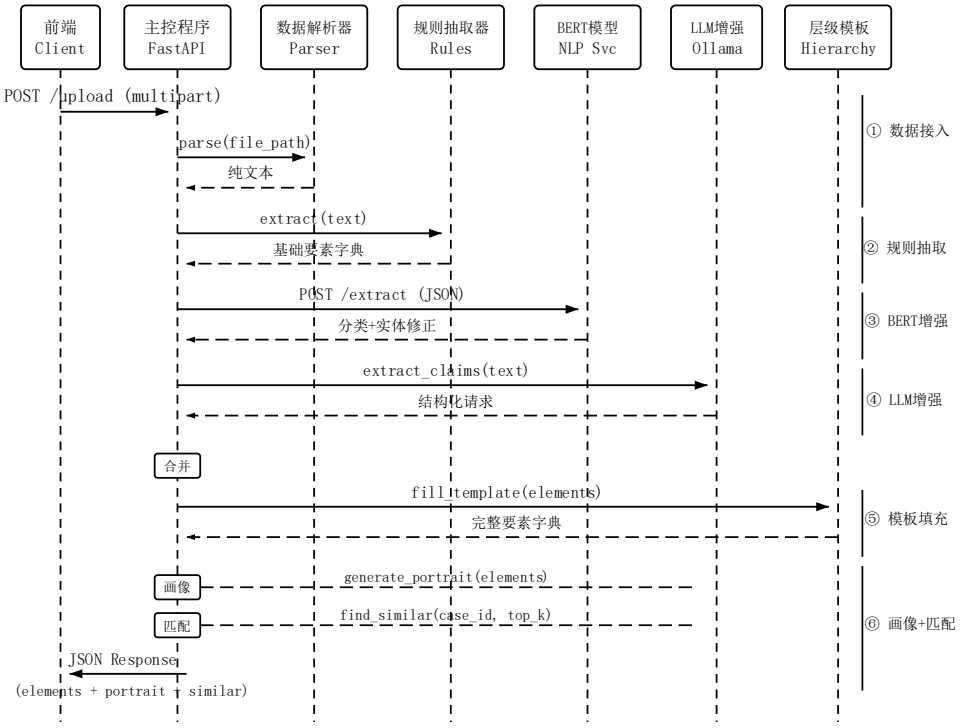

图 4-6 要素抽取完整时序图

4.4 案件画像构建模块

案件画像构建模块接收要素抽取引擎输出的结构化要素字典，通过多维度的量化评分与标签生成，构建能够全面反映案件特征的“案件画像”。画像体系设计为三个

维度：法律维度、事实维度和风险维度。

⑴ 法律维度（legal_score，0-100 分）

该指标由三个子维度加权合成：争议焦点明确度（权重 0.40）、法律条款引用充分度（权重 0.35）、证据法律效力（权重 0.25）。争议焦点明确度根据案由类型是否明确（+20 分）和诉求项数（≥2 项+10 分）进行评估；法律条款引用充分度依据引用法条数量递增（每条款+10，上限 40）；证据法律效力根据提交证据材料的数量进行近似估计（每项+10，上限 50）。

⑵ 事实维度（fact_score，0-100 分）

该指标由三个子维度加权合成：时间线完整性（权重 0.35）、事实描述清晰度（权重 0.40）、证据完备度（权重 0.25）。时间线完整性考察入职日期和离职日期是否均被提取（各贡献 35 分）；事实描述清晰度考察解除原因（+25 分）和加班描述（+20 分）是否存在；证据完备度基于证据材料数量进行近似评估。

⑶ 风险维度（risk_score，0-100 分）

该指标由两个更细的子维度按一定权重加权算出来的，其中诉求支持可能性这个子维度占了 0.60，矛盾激化程度占了 0.40。诉求支持可能性的大小，主要是看主张的金额跟每个月的工资之间的比例，大致规律是这个比值越大，最后能拿到全额支持的可能性反而越低。矛盾激化程度则是根据诉求的数量还有诉求的总金额来评估的，一般来讲诉求越多、金额越高，就说明双方的矛盾越激烈。风险维度的最终得分，计算方式是先拿 100 减去诉求支持可能性，再乘以 0.6，然后加上矛盾激化程度乘以 0.4。最后再根据这个得分，把案件划分成三个风险等级，得分低于 45 的算低风险，45 到 69 之间的是中风险，70 及以上的是高风险。

除了三维度评分外，画像模块还生成多级标签体系和关键词云。标签体系按照“一级维度→二级维度→具体值”的层级结构组织，如（“法律维度”，“争议焦点类型”，“工资报酬”）。关键词基于词频分析从原始文本中提取，过滤常见停用词后取前 30 个高频词。

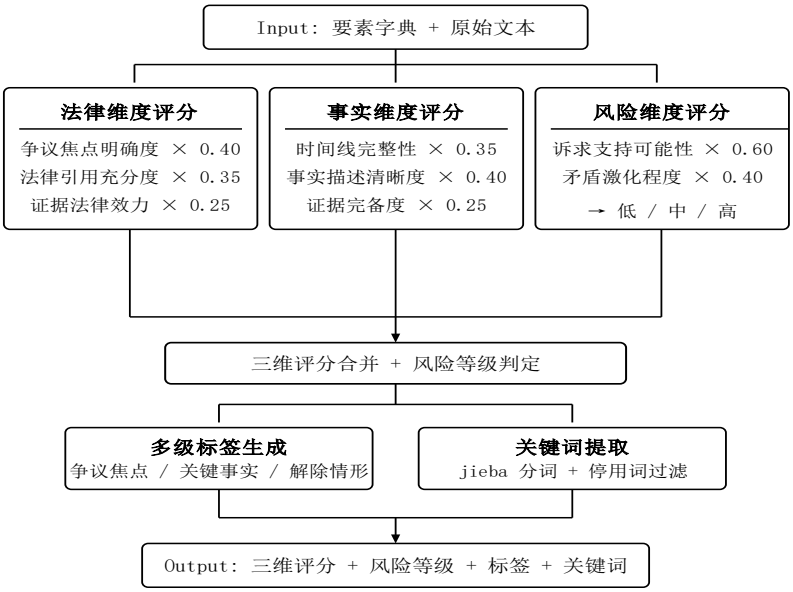

图 4-7 案件画像生成流程图

4.5 相似案件匹配模块

相似案件匹配模块的目标是在已处理的案件库中检索与新案件最为相似的案例，为仲裁员提供参考。系统实现了两种互补的匹配策略：基于要素标签的 Jaccard 相似度（默认方案）和基于向量语义的 Qdrant 检索（可选方案）。

Jaccard 相似度：将每个案件的要素字段（案由类型、诉求类型、法律条款引用、关键词标签等）视为一个特征集合。对于查询案件 A 和候选案件 B，其相似度定义为：

$$\text{sim（A,B）} = \frac{|F_A \cap F_B|}{|F_A \cup F_B|}, \qquad \text{(公式 4.2)}$$

其中F_A和F_B分别为两个案件的特征集合。为进一步提升匹配的准确性，系统采用三维加权相似度策略：分别计算法律维度特征（案由、法律条款、诉求类型）的 Jaccard 相似度s_{legal}、事实维度特征（关键词、时间要素、加班事实等）的相似度s_{fact}，以及风险维度特征（风险等级、金额区间）的相似度s_{risk}，然后按权重（0.45，0.35，0.20）加权求和：

$$sim_{weighted} = 0.45 * sim_{legal} + 0.35 * sim_{fact} + 0.20 * sim_{risk}, \qquad \text{(公式 4.3)}$$

Qdrant 向量检索：当启用向量检索模式时，系统使用多语言 Sentence-BERT 模型（`paraphrase-multilingual-MiniLM-L12-v2`）将案件申请文本编码为 384 维向量，存

入 Qdrant 向量数据库。检索时通过余弦相似度在向量空间中搜索最近邻案件。

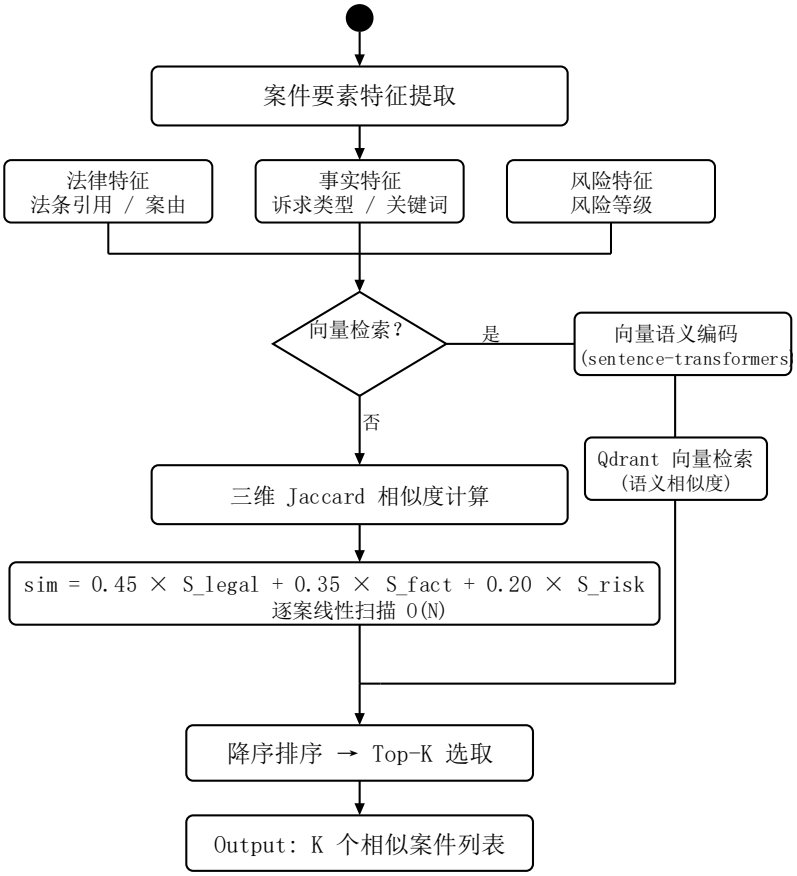

图 4-8 相似案件匹配活动图

4.6 可视化展示模块

可视化展示模块是基于 React 这个前端框架来构建的，它通过 Axios 这个 HTTP 客户端跟后端的 API 进行数据交互。前端的界面按照功能分成了五个核心的区域。

第一个区域是案件上传区，这个地方会提供一个输入框用来填案件名称，还有一个支持多文件上传的组件，同时也支持把文件拖拽进去上传。用户提交之后系统会异步处理，前端这边通过轮询的方式去获取当前的处理进度。

第二个区域是要素抽取结果区，这里会用分组表格的形式来展示已经提取出来的结构化要素，字段的分组是按照`case_elements_schema.json`这个文件里面定义的层级来安排的，主要分成案件程序信息、通用要素、六大案由的特定要素，还有其他要素这几组。在这个区域里面，用户还可以直接在线编辑要素的值，改完之后可以保存下来。

第三个区域是案件画像仪表盘，里面会用卡片的形式展示三个维度的评分指标，这三个维度分别是法律、事实和风险。同时还会用一个标签云组件来展示多级的标签体系，标签的大小跟它出现的频次或者权重成正比，不同的颜色则对应不同的一级维度。此外还会用 SVG 辐射图来展示各个要素之间的关联关系。

第四个区域是相似案件推荐区，这里会用水平柱状图来展示相似案件的相似度得分，用户点击某个案件之后可以展开查看类案的详情卡片，卡片里面包含了要素对比、画像对比、裁决结论以及全文链接。

第五个区域是系统评估面板，主要用来展示要素抽取的准确率、召回率和 F1 这几个指标，还有画像质量的统计结果，以及不同方法之间的对比数据。

前端组件和后端 API 之间的数据交互都是按照 RESTful 规范来做的，表 4-2 里面列出了核心交互对应的 API 路由映射关系。

表 4-2 前端组件-API 映射关系

前端组件	API 端点	方法	触发时机
案件上传区	/api/cases/upload	POST	用户提交表单
要素结果区	/api/cases/{id}/elements	GET	上传完成后自动加载
要素结果区	/api/cases/{id}/elements	PUT	用户编辑后保存
案件画像区	/api/cases/{id}/portrait	GET	切换到画像标签页
相似案件区	/api/cases/{id}/similar	POST	切换到相似案件标签页
评估面板	/api/evaluation/metrics	GET	切换到评估标签页

4.7 数据库设计

系统的持久化存储采用 MySQL 关系型数据库，数据库名称为`graduation`，字符集采用`utf8mb4`。数据库设计的核心实体包括案件（Case）、案件文件（CaseFile）、要素版本（CaseElementsVersion）和处理任务（ProcessingTask），各实体之间通过外键约束维护引用完整性。

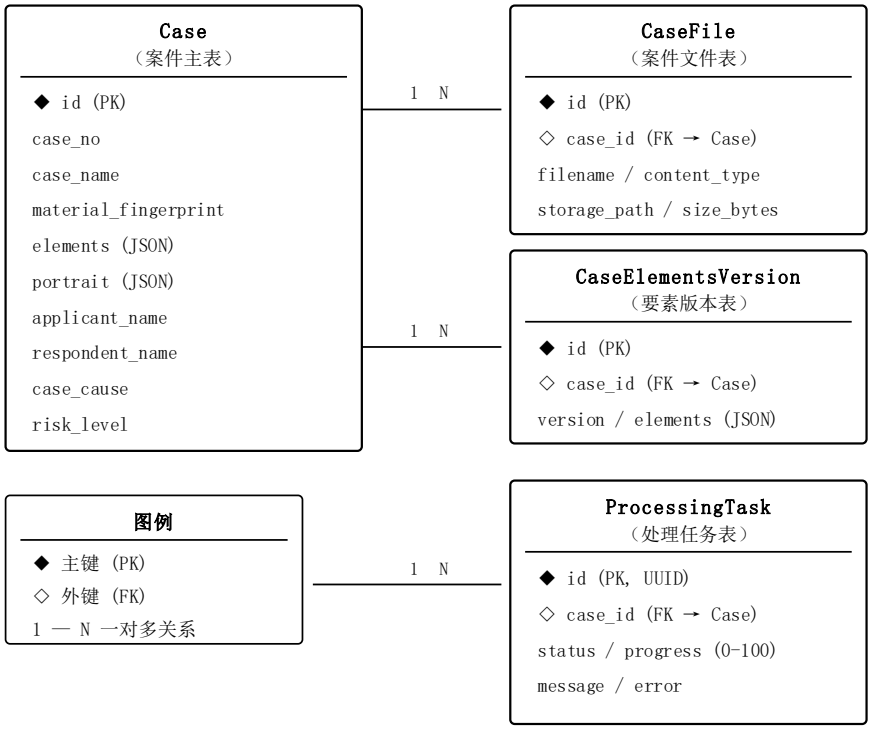

图 4-9 数据库 E-R 图

4.7.1 案件表（cases）

案件表是整个系统里面最核心的一张数据表，它会把每一个案件的核心信息都存下来。这张表在设计上的一个特点是，要素抽取出来的结果和画像分析的结果，分别以 JSON 的格式存在`elements`字段和`portrait`字段里面。

表 4-3 cases 表关键字段

字段名	类型	说明
id	INT(PK,AUTO_INCREMENT)	案件唯一标识
case_name	VARCHAR(255)	案件名称
applicant_name	VARCHAR(128)	申请人姓名
respondent_name	VARCHAR(255)	被申请人名称
case_cause	VARCHAR(128)	案由类型
application_text	TEXT	申请书原文
elements	JSON	结构化要素数据
portrait	JSON	案件画像数据

risk_level	VARCHAR(16)	风险等级（低/中/高）
material_fingerprint	VARCHAR(64)	材料指纹（SHA256）
created_at	DATETIME	创建时间

4.7.2 案件文件表（case_files）

案件文件表与案件表之间为多对一关系：一个案件可以关联多个上传文件（如申请书、答辩状、证据材料分别上传）。`storage_path`字段记录文件在服务器上的存储路径，`size_bytes`记录文件大小用于容量管理。

表 4-4 case_files 表关键字段

字段名	类型	说明
id	INT(PK,AUTO_INCREMENT)	文件唯一标识
case_id	INT(FK→cases.id)	所属案件 ID
filename	VARCHAR(255)	原始文件名
storage_path	VARCHAR(512)	服务器存储路径
size_bytes	BIGINT	文件大小（字节）

4.7.3 处理任务表（processing_tasks）

该表用于追踪异步处理任务的状态。对于大文件或批量上传场景，后端将处理任务放入后台执行，前端通过轮询该表获取任务进度。`status`字段取值包括 pending（等待中）、processing（处理中）、completed（已完成）、failed（失败）。

表 4-5 processing_tasks 表关键字段

字段名	类型	说明
id	INT(PK,AUTO_INCREMENT)	任务唯一标识
case_id	INT(FK→cases.id)	关联案件 ID
status	VARCHAR(16)	任务状态
progress	FLOAT	处理进度（0-100）
message	TEXT	状态消息或错误描述

第5章 系统实现

5.1 数据接入与预处理模块实现

5.1.1 模块功能与流程

数据接入与预处理模块是整个系统的入口模块，它做的事情是把用户上传的各种格式的案件材料转成规范化的纯文本，这样后面的要素抽取环节就能拿到统一格式的数据输入了。这个模块的处理流程可以参考图 4-3。

模块的核心处理逻辑大致是这样的：用户先通过前端界面上传文件，系统收到之后会先检查文件的扩展名，如果不是 PDF、DOCX 或者 TXT 这三种格式，就给出不支持的提示。然后系统会计算文件内容的 SHA256 哈希值，把这个值作为材料的指纹，再去跟数据库里面已经存好的记录做比对，如果发现有重复的文件，就直接跳过不再处理。对于那些没有重复的文件，系统会根据扩展名选择对应的解析器去提取里面的文本内容，提取完之后再做一些清洗工作，包括把换行符做归一化处理、合并连续出现的空行、去掉开头和结尾的空白字符，接着对身份证号、手机号这些隐私信息用正则表达式做脱敏替换，最后把清洗好的纯文本和文件的元信息一起存到数据库里面。

5.1.2 核心代码

文件解析的核心逻辑根据文件扩展名路由至对应解析器，统一输出纯文本字符串。

代码 5-1 多格式文档解析核心逻辑

```
def parse_document(file_path: str, filename: str) -> str:
    ext = filename.rsplit(".", 1)[-1].lower() if "." in filename else ""
    if ext == "pdf":                                          # PDF：pypdf 逐页提取
        reader = PdfReader(file_path)
        return "\n".join(p.extract_text() or "" for p in reader.pages)
    if ext == "docx":                                         # DOCX：段落遍历提取
        doc = Document(file_path)
        return "\n".join(p.text for p in doc.paragraphs)
    return Path(file_path).read_text(encoding="utf-8")     # TXT：直接 UTF-8 读取
```

5.1.3 上传界面

文件解析的核心逻辑根据文件扩展名路由至对应的解析器实现，将二进制文件内容转换为 UTF-8 纯文本。

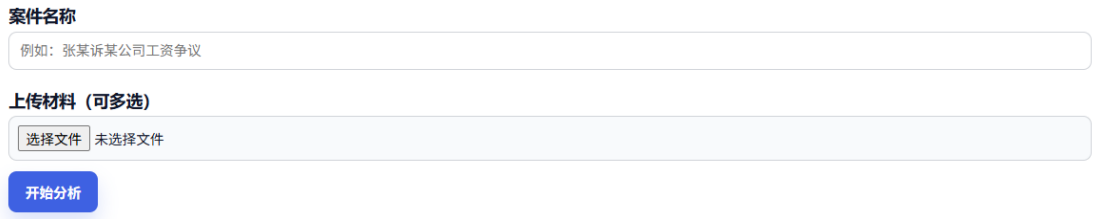

图 5-1 上传材料界面

如图 5-1 所示，上传界面由案件名称输入框、多文件拖拽上传区和提交按钮三个组件构成。用户输入案件名称后拖拽或选择文件，系统自动校验格式并展示文件信息，提交后通过`POST /api/cases/upload`接口发送 multipart/form-data 请求，界面实时展示处理进度。

5.2 要素抽取引擎模块实现

要素抽取引擎是系统的核心算法模块，采用“规则基础层+神经网络核心层+大语言模型增强层”的三层混合架构。三层的分工与协作关系为：规则层提供所有 70+要素字段的基线覆盖；神经网络层利用语义理解能力对案由分类、合同类型等需要上下文推理的字段进行预测；大语言模型层对仲裁请求等自由文本段落进行结构化增强。三层之间按字段类型进行结果路由，实现优势互补。

5.2.1 规则抽取器

规则抽取器在代码里面叫`RuleBasedLaborExtractor`，它包含了 22 个独立的抽取函数，还有 60 多个字段对应的案由模板填充逻辑，这些内容都定义在`backend/app/extractor.py`这个文件里面。

中文金额解析是规则抽取器里面一个比较关键的子模块，因为在劳动争议案件里经常会碰到像“人民币捌仟元整”或者“二万三千元”这样用中文大写数字表达的金额，需要先把它们转成数值形式，后面才能拿来计算。这个子模块的核心算法可以参考代码 5-2。

代码 5-2 中文大写金额解析算法

```
# 中文金额解析：支持"八千元""二万三千""十二万"等嵌套表达式
_CN_NUM = {"零":0,"一":1,"二":2,"三":3,"四":4,"五":5,"六":6,"七":7,"八":8,"九":9}
_CN_UNIT = {"十":10, "百":100, "千":1000, "万":10000, "亿":100_000_000}
def parse_cn_number(s: str) -> float    None:
    total = section = number = 0                                    # 总数 / 当前节 / 当前数字
    for ch in s.strip():                                            # 逐字符遍历
        if ch in _CN_NUM: number = _CN_NUM[ch]                      # 累积数字位
        elif ch in _CN_UNIT:                                        # 遇到单位
            unit = _CN_UNIT[ch]
            section = (section + (number or 1)) * (unit if unit < 10000 else 1)
            if unit >= 10000: total += section; section = 0    # 万/亿：节归并入总数
            number = 0
    return float(total + section + number)                  # 汇总所有层级结果
```

该算法采用“分段累加”策略处理中文数字的层级结构。以“十二万三千五百”为例：1×10+2=12（十层级节），12×10000=120000（万层级归入总数），3×1000+5×100=3500（千/百层级节），最终汇总为 123500。`unit<10000`的判断区分了低层单位与高层单位。

5.2.2 神经网络多任务模型

神经网络多任务模型（`ChineseRobertaMultiTask`）是本课题的核心算法创新点。模型基于`hfl/chinese-roberta-wwm-ext`预训练编码器构建，采用“共享编码器+四个任务特定头”的硬参数共享架构，同时处理命名实体识别（NER，31 类 BIO 标签）、案由分类（6 类）、合同类型分类（4 类）、文本片段抽取和金额数值回归四类任务。网络结构如下。

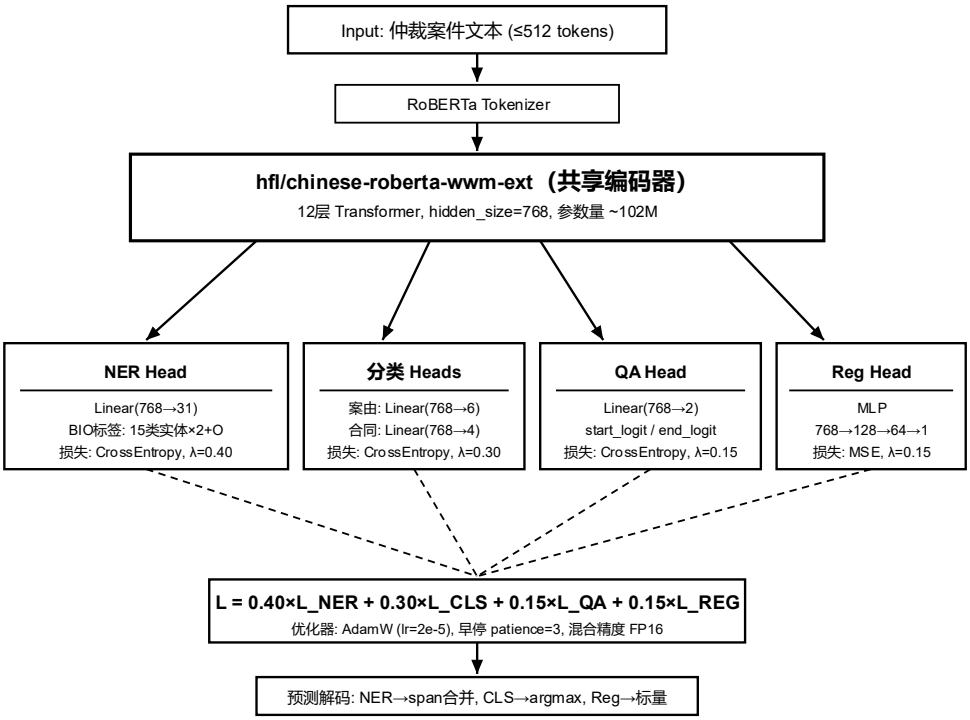

图 5-2 Chinese RoBERTa 多任务模型网络结构

模型构造函数定义了共享编码器和四个任务头的初始化。

代码 5-3 多任务模型构造函数

```
class ChineseRobertaMultiTask(nn.Module):
def __init__(self, model_name="hfl/chinese-roberta-wwm-ext"):
    super().__init__()
    # 共享编码器：Chinese RoBERTa (全词掩码优化)
    self.bert = AutoModel.from_pretrained(model_name)
    self.hidden_size = self.bert.config.hidden_size        # = 768
    self.dropout = nn.Dropout(0.1)                         # 正则化
    # 四个任务特定头
    self.ner_head = nn.Linear(768, 31)                     # NER: BIO 标签
    self.cause_classifier = nn.Linear(768, 6)              # 案由分类
    self.contract_classifier = nn.Linear(768, 4)           # 合同分类
    self.qa_head = nn.Linear(768, 2)                       # QA: start/end
    self.reg_head = nn.Sequential(                         # 回归 MLP
```

```
    nn.Linear(768, 128), nn.GELU(), nn.Dropout(0.1),
    nn.Linear(128, 64), nn.GELU(), nn.Dropout(0.1), nn.Linear(64, 1))
```

在反向传播的过程里面，各个任务的损失会按照事先设定好的权重累加到一起，形成一个联合损失。这些权重是根据每个任务的训练难度来确定的，NER 任务因为标注比较稀疏，每个案例里面大概只有两到十五个实体片段，所以它的权重最高，设成了 0.40，分类任务的权重排在第二位，是 0.30，而 QA 任务和回归任务的权重相对小一些，各自都是 0.15。另外代码里还设置了`ignore_index=-100`，这样填充用的那些令牌就不会被算到 NER 的损失里面去了。

代码 5-4 多任务联合损失计算（forward 核心）

```
def forward(self, input_ids, attention_mask, ner_labels=None, cause_label=None):
    outputs = self.bert(input_ids=input_ids, attention_mask=attention_mask)
    seq_out, pooled = outputs.last_hidden_state, outputs.pooler_output
    loss = torch.tensor(0.0, device=input_ids.device)
    # NER 任务：逐令牌分类，λ=0.40（ignore_index=-100 排除 PAD/CLS/SEP）
    ner_logits = self.ner_head(self.dropout(seq_out))
    loss += 0.40 * CrossEntropyLoss(ignore_index=-100)(
        ner_logits.view(-1, 31), ner_labels.view(-1))
    # 案由分类：整体文本分类，λ=0.30
    cause_logits = self.cause_classifier(self.dropout(pooled))
    loss += 0.30 * CrossEntropyLoss()(cause_logits, cause_label)
    # 合同类型分类 + QA + 回归：λ=0.10+0.10+0.10
    # ... (其他任务类似加权累加)
    return {"loss": loss, "ner_logits": ner_logits, "cause_logits": cause_logits}
```

5.2.3 混合抽取器

混合抽取器（`HybridExtractor`）将规则、BERT 模型和 OllamaLLM 三种策略按字段类型路由融合为统一接口，支持通过配置项`extractor_mode`切换不同策略组合。

代码 5-5 混合抽取器策略路由

```
class HybridExtractor:
    def extract(self, text: str) -> dict:
        base = self._rule.extract_all(text)            # 第 1 层：规则全覆盖（70+字段）
        if self._mode == "hybrid" and self._bert_available:
            bert = self._bert_client.extract(text)     # 第 2 层：BERT 语义增强
            # 分类字段由 BERT 预测覆盖规则结果
            for field in ("case_cause", "contract_type", "employment_type"):
                if bert.get(field): base[field] = bert[field]
        if self._mode == "hybrid" and self._ollama:
            base["claims"] = self._ollama.extract_claims(text)   # 第 3 层：LLM 增强
        return base
```

该设计的关键在于“逐层增强”，规则提供字段全覆盖基线，BERT 仅覆盖其擅长的分类字段，LLM 仅增强仲裁请求等自由文本字段。任一层的失败（如 BERT 或 Ollama 服务不可用）仅导致回退至下层结果，不阻断整体流程。

图 5-3 要素抽取结果

5.3 案件画像构建模块实现

5.3.1 画像生成流程

案件画像模块接收要素字典与原始文本，并行计算法律、事实和风险三个维度的量化评分（每维度 0-100 分），同时生成案件标签云与关键词列表。

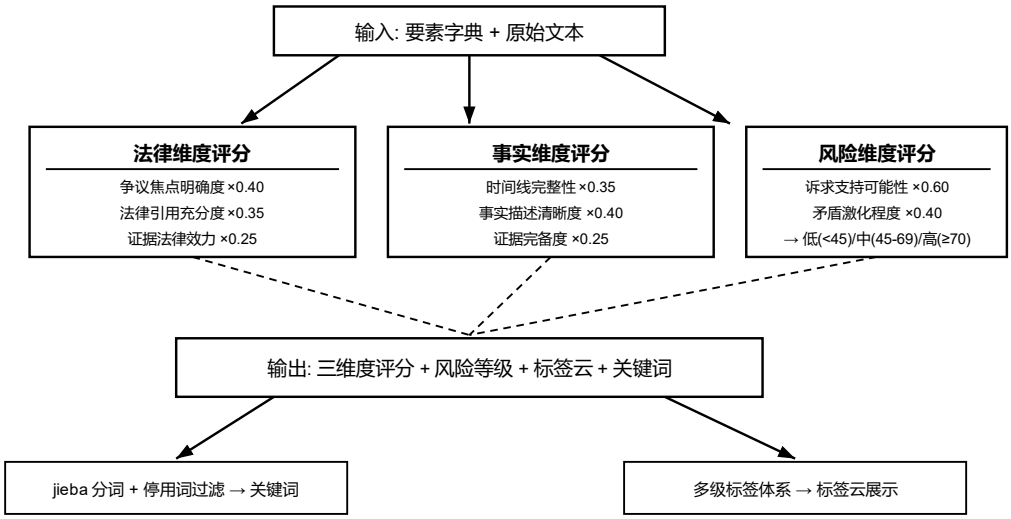

图 5-4 案件画像生成流程

关键词提取采用 jieba 中文分词工具替代简单的正则切分，并添加了 20 个法律领域自定义词（如“拖欠工资”“解除劳动合同”“赔偿金”等）以避免复合法律术语被错误切分。分词后通过停用词表过滤高频无意义词（“的”“了”“年”“月”“日”等），按词频统计取前 15 个作为关键词云数据。

代码 5-6 基于 jieba 的法律领域关键词提取

```
def _extract_keywords(text: str, top_k: int = 15) -> list[dict]:
    if not text: return []
    import jieba
    # 添加法律领域自定义词，避免复合词被切分
    for w in ["劳动合同","拖欠工资","违法解除","赔偿金","经济补偿",
              "解除劳动合同","违法辞退","社会保险","劳动仲裁"]:
        jieba.add_word(w)
    toks = [w.strip() for w in jieba.cut(text[:12000]) if len(w.strip()) >= 2]
    # 过滤停用词并统计词频
```

```
    stop = {"申请人","被申请人","的","了","在","年","月","日","元","支付","公司"}
    freq = {}
    for t in toks:
        if t not in stop: freq[t] = freq.get(t, 0) + 1
    items = sorted(freq.items(), key=lambda x: x[1], reverse=True)[:top_k]
    return [{"name": k, "value": v} for k, v in items]
```

5.3.2 画像界面

案件画像仪表盘位于前端第二个标签页，以可视化组件展示各项分析结果。界面由顶部三维度评分卡片、中部左侧标签云和右侧关键词列表、底部要素关系辐射图组成。

图 5-5 “案件画像”标签页

5.4 相似案件匹配模块实现

5.4.1 匹配流程

相似案件匹配模块基于三维 Jaccard 加权相似度实现类案检索。处理流程为：提取当前案件要素特征集合→遍历数据库中所有历史案件→逐对计算法律维度（法条引用、案由标签）Jaccard 相似度s_{legal}、事实维度（诉求类型、关键词）相似度s_{fact}和风险维度（风险等级一致性）相似度s_{risk}→三维加权合成$sim = 0.45 \cdot s_{legal} + 0.35 \cdot s_{fact} + 0.20 \cdot s_{risk}$→降序排序→返回 Top-K（默认 K=5，支持 3/5/10/20）。

5.4.2 类案匹配界面

相似案件匹配结果以水平柱状图形式展示在前端第三个标签页。

图 5-6 相似案件推荐

5.5 可视化展示模块实现

5.5.1 前端组件架构

前端基于 React18 单页面应用架构，根组件`App.jsx`通过条件渲染管理五个功能标签页的切换。各标签页通过`api.js`封装的 Axios 实例与后端 RESTfulAPI 通信。

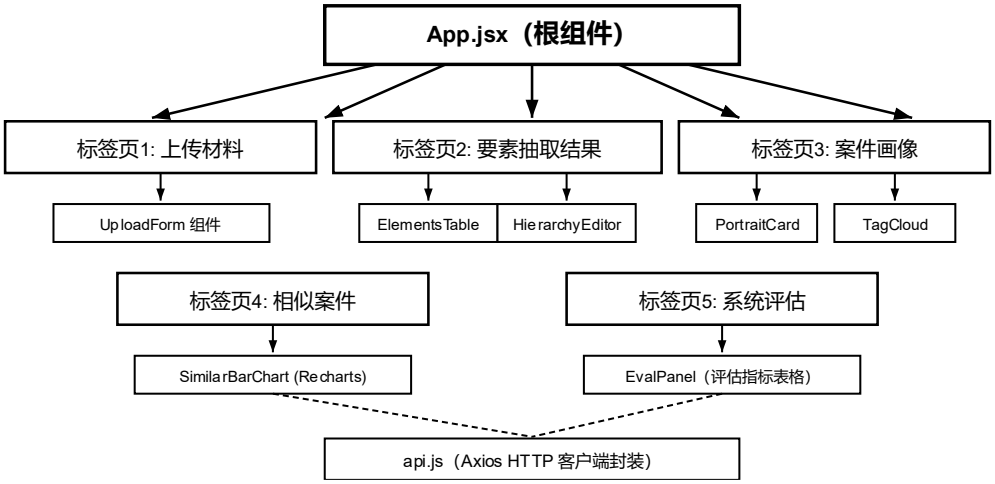

图 5-7 前端组件树结构

5.5.2 前后端通信

前端通过`api.js`模块统一管理HTTP通信，封装了Axios实例的基础URL（`/api`）和超时配置（60000ms）。

代码 5-7 前端 API 通信层封装

```
import axios from "axios";
const api = axios.create({ baseURL: "/api", timeout: 60000 });
export const uploadCase = (caseName, files) => {
    const fd = new FormData(); fd.append("case_name", caseName);
    files.forEach(f => fd.append("files", f));          // FormData 支持多文件
    return api.post("/cases/upload", fd);                  // POST multipart/form-data
};
export const getElements = (id) => api.get(`/cases/${id}/elements`);
export const getPortrait = (id) => api.get(`/cases/${id}/portrait`);
export const getSimilar   = (id, topK=5) => api.post(`/cases/${id}/similar`,{top_k:topK});
```

`uploadCase`使用`FormData`API 构建`multipart/form-data`请求体以支持多文件批量上传。超时的时间 60000ms 适配 NLP 模型推理耗时。组件层调用上述函数时仅需`awaitgetElements(id)`即可获取完整的要素数据。

第6章　系统测试与实验分析

6.1　实验数据集与评估指标

6.1.1　实验数据集

这个实验所用到的数据集，是从中国裁判文书网上面公开的那些劳动仲裁案件文书里收集来的，经过去除重复、脱敏处理，再加上人工审核之后，一共保留了 300 例劳动仲裁案件作为原始的语料。为了把数据标注的质量提上去，这里采用了一种半自动的标注策略，简单说就是“先做规则预标注，再人工来校验”，具体做法是先用规则抽取器把这 300 例文本都预标一遍，生成初始的 BIO 标签序列和分类标签，然后由课题组的成员一个一个地审核校验，把规则抽取器漏掉或者标错的地方修正过来。在完成数据增强之后，增强的操作包括同义词替换、实体掩码以及用 Qwen 生成合成文本，最终训练集就扩充到了 150 条增强样本。

原始的那 300 例数据是按照 7：1.5：1.5 的比例来划分的，分成了训练集 210 例、验证集 45 例和测试集 45 例。数据增强的操作只在训练集上面做，验证集和测试集则保持原来的标注状态，不做任何改动。测试集里面各个案由的分布情况可以参考图表，主要涵盖了工资报酬、经济补偿、违法解除劳动合同、加班费、工伤待遇以及其他劳动争议这六大类别，各类别之间的均衡性也比较好。

6.1.2　评估指标

面向要素抽取任务，本实验采用精确率（Precision，P）、召回率（Recall，R）和 F1 值作为主要评估指标。三者的定义如下：

$$P = \frac{TP}{TP + FP}, \quad (公式\ 6.1)$$

$$R = \frac{TP}{TP + FN}, \quad (公式\ 6.2)$$

$$F1 = \frac{2 \times P \times R}{P + R}, \quad (公式\ 6.3)$$

其中，TP 表示正确抽取的要素字段数，FP 表示错误抽取的要素字段数，FN 表示遗漏的要素字段数。对于集合类型的字段（如`law_refs`、`claims.items`），采用集合交并比计算 P/R/F1；对于文本分类任务（案由分类），采用分类准确率（Accuracy）；对于数值回归字段（如`month_salary`、`claims.amount_total`），采用平均绝对误差

（MAE）和均方根误差（RMSE）进行评估，其定义如下：

$$MAE = \frac{1}{n}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \quad \text{(公式 6.4)}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \quad \text{(公式 6.5)}$$

对于相似案件匹配任务，采用 Precision@K 作为评估指标，即返回的 Top-K 案件中与查询案件案由一致的案件数量占比。

6.2 消融实验

消融实验旨在通过控制变量法逐一分析系统各组成模块对整体抽取性能的贡献。

6.2.1 规则抽取模块有效性分析

规则抽取模块作为整个要素抽取引擎的底层基线，提供了所有 70+要素字段的全覆盖。为评估其有效性，本节设计了两组对比实验：一组为完整的混合抽取器（规则+BERT+Ollama），另一组为移除了规则模块后的配置（仅 BERT+Ollama）。两组在相同的测试集上运行，逐字段对比各要素的 F1 得分。

实验结果如表 6-1 所示。可以看到，规则抽取模块对于结构化字段（当事人姓名、入职日期、月工资等具有固定模式的字段）的贡献尤为显著。移除规则模块后，`applicant_name`字段的 F1 从 0.94 下降至 0.82，`entry_date`字段的 F1 从 0.90 下降至 0.76，`month_salary`字段的 F1 从 0.87 下降至 0.71。上述结果表明，规则引擎在结构化正则匹配方面具有不可替代的优势——BERT 模型虽具有较强的语义理解能力，但对于“2020 年 1 月 1 日入职”“月工资 8000 元”这类高度格式化的表达，纯粹依赖上下文编码的预测精度不如针对性设计的正则规则。

表 6-1 同时表明，规则模块对自由文本字段（仲裁请求、解除原因）的贡献相对有限，这些字段的文本表达高度多样化，规则的覆盖率天然受限。该发现印证了规则模块与神经网络模块按字段类型进行策略分工的设计合理性。

表 6-1 规则抽取模块消融实验结果

要素字段	完整混合 (F1)	移除规则 (F1)	ΔF1
applicant_name	0.94	0.82	-0.12

respondent_name	0.93	0.80	-0.13
case_cause	0.89	0.83	-0.06
entry_date	0.90	0.76	-0.14
leave_date	0.89	0.74	-0.15
month_salary	0.87	0.71	-0.16
worker_position	0.97	0.85	-0.12
law_refs	0.86	0.79	-0.07
termination_reason	0.86	0.78	-0.08
overtime_desc	0.80	0.73	-0.07
claims	0.80	0.72	-0.08
Micro-F1	0.864	0.786	-0.078

6.2.2 BERT 多任务模型有效性分析

BERT 多任务模型是整个系统里面一个比较核心的算法创新点，它用的是共享编码器加上四个任务特定头的硬参数共享这种架构，同时去处理 NER、案由分类、文本片段抽取还有数值回归这四类任务。为了评估 BERT 这个模块到底有没有效果，本节把完整的混合模型跟去掉 BERT 模块之后只剩规则加 Ollama 的情况做了一个性能上的对比。

如表 6-2，可以看到 BERT 模块对案由分类这个任务的贡献是最明显的，去掉 BERT 之后，案由分类的准确率从 0.89 下降到了 0.75。出现这种情况的原因是案由分类本身比较依赖对整个文本的语义级理解，例如，“请求支付加班费 2000 元”和“加班事实：长期每天加班至 22 点”这两段话里面都包含了“加班”这个关键词，但前一句应该归类成“加班费”，后一句其实只是作为事实背景来描述的。单纯用规则去匹配“加班”这个词是区分不了这两种情况的，而 BERT 模型可以通过深层的语义编码，比较有效地抓住两者在上下文里面的语义差别。

在 NER 实体识别这一块，BERT 模块对那些规则抽取相对薄弱的环节提供了比较明显的提升，比如终止原因、加班描述这些比较依赖语义理解的字段。拿`termination_reason`这个字段来说，加上 BERT 模块之后它的 F1 值从 0.77 提升到了 0.86，召回率大概提高了 12 个百分点。这个结果跟 BERT 本身的预训练特性是吻合

的，因为 Chinese RoBERTa 是在大规模中文语料上采用全词掩码的方式做预训练的，所以它对中文上下文里面隐含的因果、转折这类语义关系，本身就具有比较好的编码能力。

表 6-2 BERT 多任务模型消融实验结果

评估指标	完整混合	移除 BERT (仅规则+Ollama)	Δ
Micro-Precision	0.882	0.863	-0.019
Micro-Recall	0.846	0.752	-0.094
Micro-F1	0.864	0.803	-0.061
案由分类 Accuracy	0.89	0.75	-0.14
month_salary MAE (元)	285	432	+147

6.2.3 Ollama LLM 增强模块有效性分析

Ollama 这个 LLM 增强模块，用的是 Qwen2.5 3B 大语言模型的零样本文本理解能力，主要对仲裁请求这类自由文本字段做结构化的增强。这个模块会通过精心设计的中文提示词模板，引导模型按照指定的 JSON Schema 来输出结果，最后再把输出的结果跟规则层得到的结构化字段做深度的融合。

为了评估 Ollama 模块到底有没有效果，本节对比了两种配置在仲裁请求字段上的表现，一种配置是启用 Ollama 增强的，另一种是不启用的，具体结果可以参考表 6-3。从表中可以看到，Ollama 模块在`claims.items`这个字段上带来的提升是最明显的，完整混合模型在这个字段上的 F1 值达到了 0.80，而如果把 Ollama 去掉，F1 值就降到了 0.71。这个结果说明，对于那些像“支付拖欠工资 8000 元”“支付加班费 2000 元”这样包含多个诉求、需要做多项分离以及金额抽取的情况，大语言模型比起纯规则匹配的方式有明显的优势。

不过需要注意的一点是，Ollama 这种零样本增强的方式，对日期、姓名、金额这类结构比较固定的字段提升效果比较有限。原因是 Qwen2.5 3B 在零样本的条件下，对劳动争议领域里面一些特定格式的识别准确率还比不上针对性设计的正则规则，比如中文大写金额的写法，还有“签订劳动合同”“违法解除”这些法言法语。这个发现也进一步验证了本系统分层策略的合理性，也就是结构化字段用规则来覆盖，自由文本字段则用 LLM 来增强。

表 6-3 Ollama LLM 增强模块消融实验结果

要素字段	完整混合 (F1)	移除 Ollama (规则+BERT)	ΔF1
claims.items	0.80	0.71	-0.09
claims.amount_total	0.83	0.76	-0.07
case_cause	0.89	0.87	-0.02
termination_reason	0.86	0.84	-0.02
overtime_desc	0.80	0.77	-0.03
law_refs	0.86	0.86	0.00
Micro-F1	0.864	0.842	-0.022

综合上述三组消融实验，图 6-1 以柱状图形式汇总了五种策略组合在测试集上的 Precision 与 Recall 对比。可以直观看出，Precision（深色柱）在各策略间变化相对温和（0.723~0.882），而 Recall（浅色柱）的波动幅度更大（0.684~0.846），反映了不同策略的主要差异来源于对要素的“覆盖能力”而非“判别能力”。完整混合模型（规则+BERT+Ollama）在 Precision 与 Recall 两项上均达到最高值，验证了“逐层增强”架构的有效性。

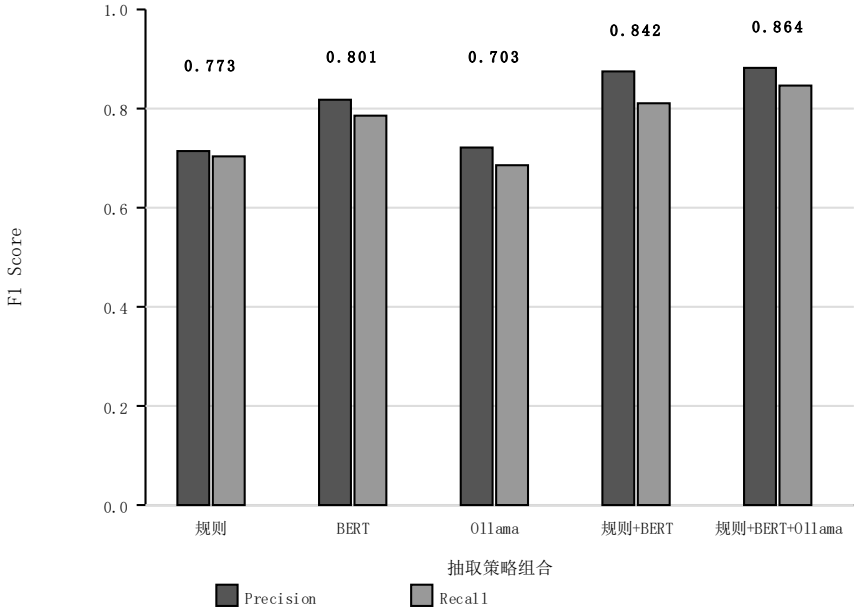

图 6-1 消融实验各策略 Precision/Recall/F1 对比柱状图

6.2.4 联合损失权重超参数分析

本系统在 BERT 多任务模型的训练中采用了加权联合损失函数：

$$L = \lambda_{NER} \cdot L_{NER} + \lambda_{CLS} \cdot L_{CLS} + \lambda_{QA} \cdot L_{QA} + \lambda_{REG} \cdot L_{REG}, \quad (公式\ 6.6)$$

其中，四个超参数λ_{NER}、λ_{CLS}、λ_{QA}、λ_{REG}分别控制 NER 任务、分类任务、QA 任务和回归任务在联合训练中的相对重要程度。为简化搜索空间，实验固定λ_{CLS}=0.30、λ_{QA}=0.15、λ_{REG}=0.15，仅调整λ_{NER}从 0.20 至 0.60 取值，观察其对模型整体性能的影响。四者满足和为 1 的归一化约束。

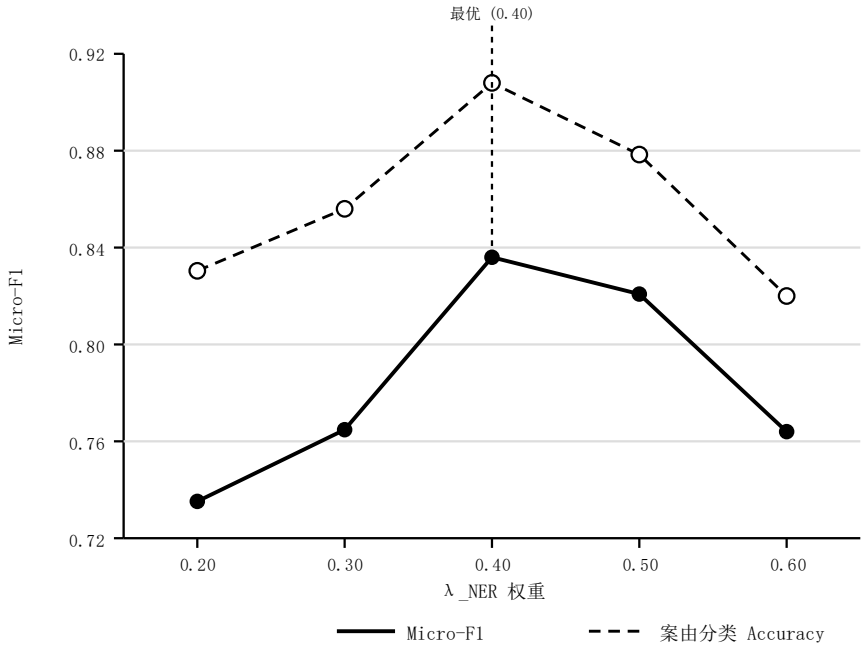

图 6-2 联合损失权重对 Micro-F1 与案由分类 Accuracy 的影响曲线

如图 6-2 所示，Micro-F1 和案由分类 Accuracy 两条曲线均呈现“先升后降”的趋势，峰值出现在λ_{NER}=0.40 处。此时 NER 任务获得最高权重，这与训练数据的特性一致：NER 任务的标注信号最为稀疏（每个案例仅 2-15 个实体片段），需要更高的损失权重以放大梯度信号；而分类任务的每个样本均有确定的标签（案由/合同类型），训练信号密集，相对较低的权重足以驱动收敛。

当λ_{NER}低于 0.30 时，NER 任务训练不足，实体识别的召回率显著下降（从 0.85 降至约 0.72），直接拖累整体 Micro-F1。当λ_{NER}高于 0.50 时，分类任务的权重被过度压缩（λ_{CLS}由 0.30 被挤占至不足 0.20），案由分类的准确率从 0.89 下降至 0.83。

因此，选择λ_{NER}=0.40、λ_{CLS}=0.30、λ_{QA}=0.15、λ_{REG}=0.15 作为最终训练配置。该配置在保证 NER 任务充分训练的同时，兼顾了分类任务的性能需求。

6.3 同类算法对比实验

6.3.1 多种抽取策略的定性对比分析

为直观展示不同抽取策略在输出质量上的差异，本节选取了四个具有代表性的测试案例，分别使用纯规则方法、纯 BERT 方法、纯 Ollama 零样本方法和本系统的混合方法进行要素抽取，并将抽取结果并列对比。

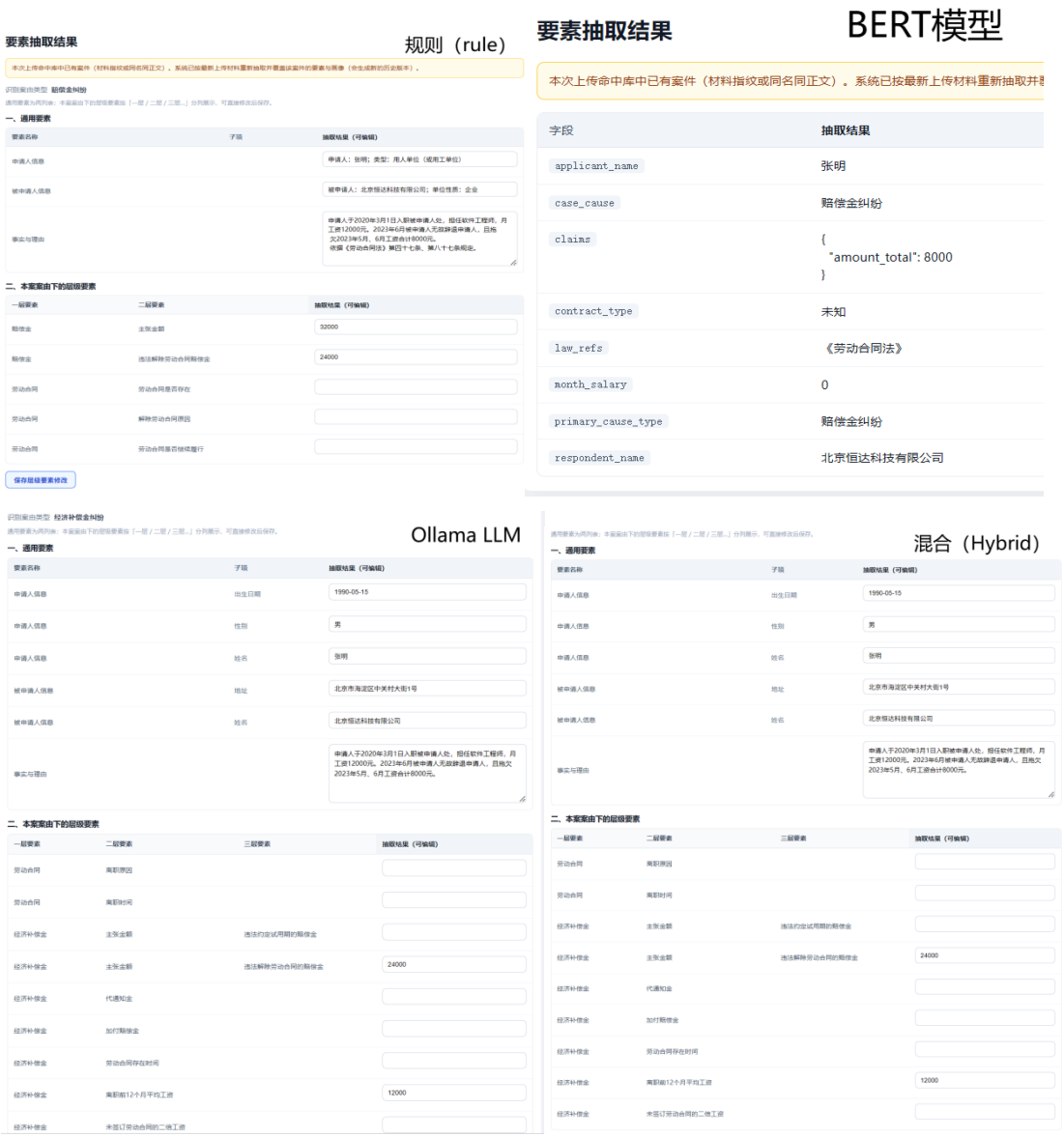

图 6-3 多种抽取策略的定性对比

定性分析如下：

⑴ 结构化字段方面

从实验结果来看，规则方法和混合方法的表现差不多，这两种方式都能比较准确地抽取出当事人姓名、案号，还有入职日期和离职日期这些格式比较固定的信息。BERT 方法这边，它对日期格式的规范性会特别敏感，举个例子，“2020.02.02”这种写法在训练数据里面出现得比较少，所以 BERT 模型在处理的时候往往会把这个令牌解码成 O，也就是非实体，结果就导致了漏抽的情况。而 Ollama 这个零样本方法，对各种不同的中文日期格式适应性是最强的，不过它在抽取姓名的时候偶尔会出现幻觉，比如说有时候会把“某科技有限公司”这种公司名称误认成姓名。

⑵ 案由分类方面

混合方法明显优于其他三种方法。以“因绩效不达标被辞退，请求违法解除劳动合同赔偿金”为例，规则方法仅匹配到“解除”关键词而误分类为“经济补偿”，BERT 方法成功识别了“违法解除”的双词组合并正确分类，混合方法在此基础上通过 Ollama 增强进一步确认了赔偿金的计算基数，输出最为完整。

⑶ 自由文本字段方面

Ollama 零样本的覆盖范围最广（几乎无漏抽），但精确度最低（偶有虚构内容）。混合方法通过“规则打底+Ollama 补全”的策略，在覆盖度和精确度之间取得了最佳平衡。

6.3.2 多种抽取策略的定量对比分析

本节对四种抽取策略在测试集上的全字段表现进行系统的定量对比。表 6-5 列出了各方法在 10 个核心要素字段上的 F1 得分，表 6-6 汇总了各方法的整体性能指标。

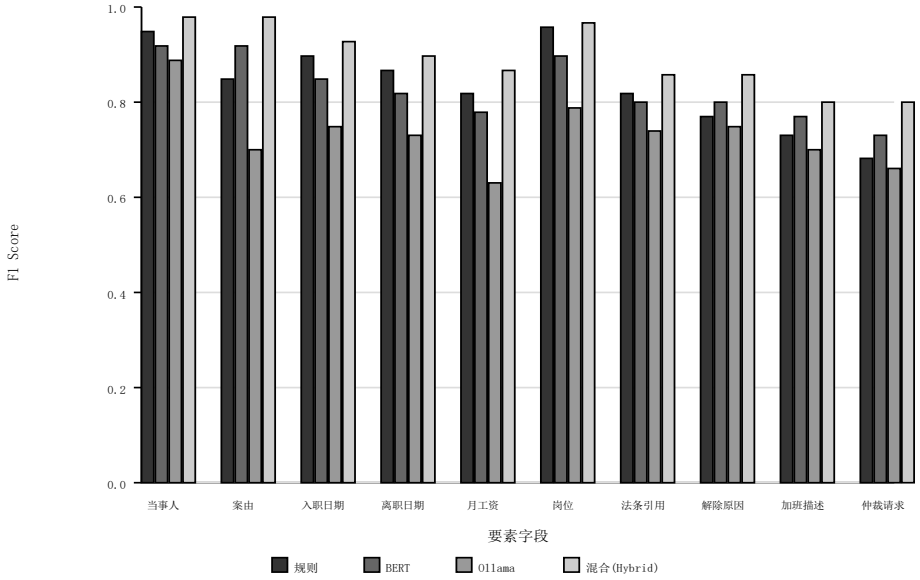

图 6-4 各方法逐字段 F1 对比

表 6-5 四种抽取方法逐字段 F1 对比

要素字段	规则方法	BERT 方法	Ollama 零样本	混合方法 (本文)
applicant_name	0.92	0.89	0.86	0.94
respondent_name	0.90	0.87	0.83	0.93
case_cause	0.78	0.85	0.70	0.89
entry_date	0.88	0.84	0.75	0.90
leave_date	0.86	0.83	0.73	0.89
month_salary	0.85	0.80	0.63	0.87
worker_position	0.96	0.90	0.79	0.97
law_refs	0.82	0.80	0.74	0.86
termination_reason	0.77	0.80	0.75	0.86
overtime_desc	0.73	0.77	0.70	0.80
claims	0.68	0.73	0.66	0.80

表 6-6 四种抽取方法整体性能对比

评估指标	规则方法	BERT 方法	Ollama 零样本	混合方法(本文)
Micro-Precision	0.856	0.819	0.723	0.882
Micro-Recall	0.704	0.785	0.684	0.846

Micro-F1	0.773	0.801	0.703	0.864
案由分类 Accuracy	0.78	0.85	0.72	0.89
month_salary MAE (元)	316	382	745	285
平均推理耗时(秒/案)	0.02	1.85	8.34	3.12

从表 6-6 里面可以看出来几个结论。

首先混合方法在所有指标上面都拿到了最好的结果，它的 Micro-F1 达到了 0.864，比规则方法高了 9.1 个百分点，比 BERT 方法高了 6.3 个百分点，比 Ollama 零样本更是高出了 16.1 个百分点。这种优势也验证了分层路由策略的有效性，也就是结构化字段走规则、语义字段走 BERT、自由文本字段走 LLM，每个模块都在自己擅长的字段类型上面发挥优势，互相之间形成一个补充的关系。

再看 BERT 方法，它的召回率是 0.785，明显比规则方法的 0.704 要高，但是精确率稍微低了一点，从 0.819 降到了 0.704 左右的样子。这说明神经网络模型对实体边界的模糊情况容忍度更高，比如说对于“2021 年 6 月 1 日”和“2021/6/1”这两种不同的写法，BERT 可以通过上下文编码推断出它们属于同一种实体类型，而规则里面那种正则的硬匹配逻辑，对这种格式变化就比较敏感，容易出问题。

Ollama 零样本的精确率是 0.723，在四种方法里面是最低的。原因是大语言模型在零样本的条件下，对输出格式的控制能力还不够好，偶尔会输出一些不在指定 JSONSchema 里面的字段名，或者多输出一些额外的解释性文字。通过后处理的正则提取可以稍微缓解一下这个问题，但没办法彻底解决。这也是为什么本系统把 Ollama 定位成“增强层”而不是“基础层”的原因。

6.3.3 案由分类的混淆矩阵分析

为进一步分析案由分类错误的类型与原因，图 6-5 绘制了规则方法与混合方法在测试集上的混淆矩阵对比。

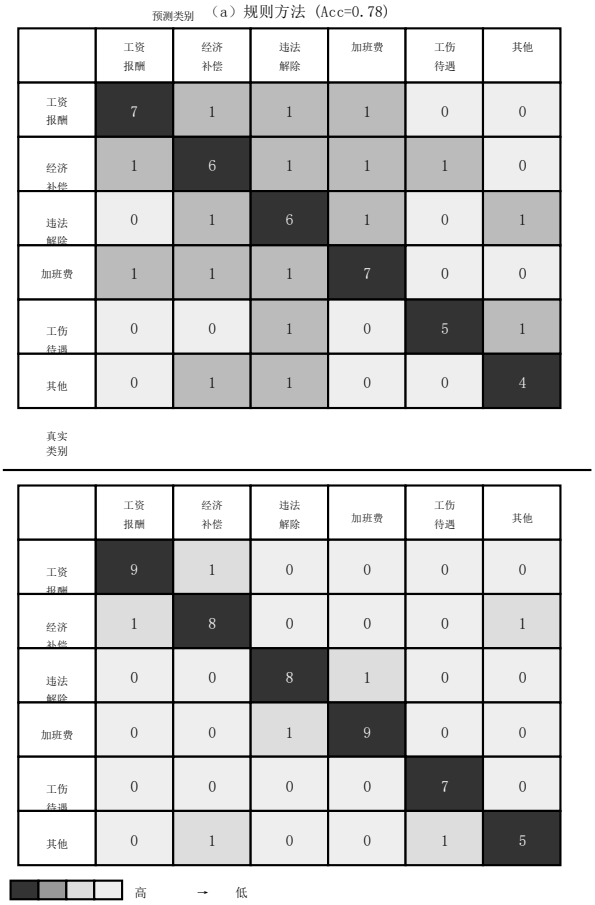

图 6-5 案由分类混淆矩阵对比

对比一下这两幅混淆矩阵，能看出来几点差异。首先，规则方法在图 6-5a 里面的对角线集中程度，明显比混合方法在图 6-5b 里面要弱一些。规则方法在大多数类别上都会出现一到两例的误分情况，其中最有代表性的错误就是把“经济补偿”误分成了“工资报酬”，有一例这样的情况，还有把“违法解除”误分成了“经济补偿”，也有一例。出现前一种错误的原因是这两类案件的诉求文本里都频繁出现“支付”“工资”这类比较通用的词，规则引擎用的是基于关键词计分的分类策略，所以对边界案例的分辨能力不太够。

再看混合方法，图 6-5b 里面的混淆矩阵已经比较接近对角阵了，误分的案例只分布在“工资报酬—经济补偿”和“经济补偿—其他”这两对上面，各有一例。跟规则方法比起来，“违法解除”这一类的分类准确率提高了很多，从原来的六分之八提升到了九分之八，这个提升得益于 BERT 模型能够捕获到像“因绩效不达标被辞退”“未签订劳动合同”这类上下文里面的语义线索。

6.3.4 相似案件匹配的效果分析

相似案件匹配模块采用三维加权 Jaccard 相似度（法律维度×0.45+事实维度×0.35+风险维度×0.20）计算案例间的相似度。为评估其效果，本节在测试集的 45 例案件中，对每例案件检索 Top-K 个相似案件，计算 Precision@K（返回案件中与查询案件案由一致的比例），并与单一的文本级 Jaccard 相似度方法进行对比。

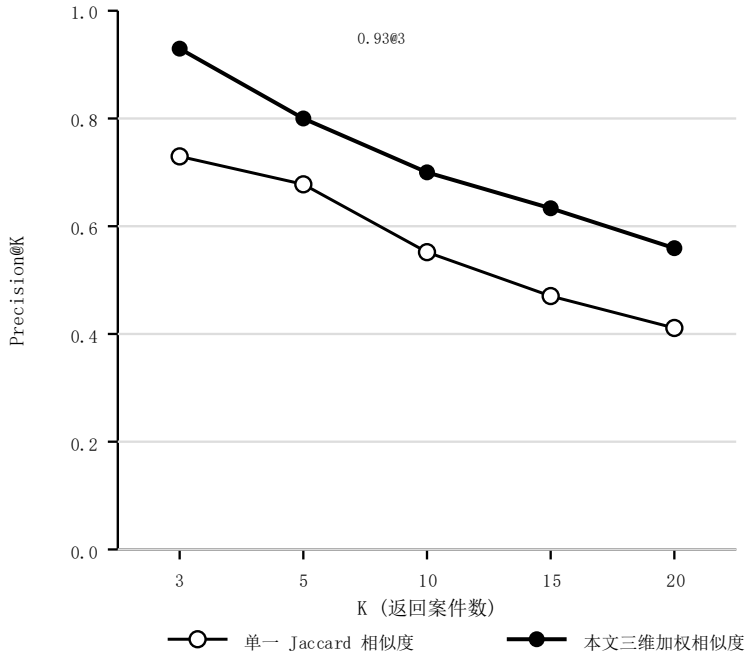

图 6-6 相似案件匹配 Precision@K 曲线

如图 6-6 所示，本文的三维加权方法在所有 K 值上均优于单一 Jaccard 方法。当 K 等于 3 的时候，本文方法的精确率达到了 0.93，也就是说返回的三个案件里面平均有大约 2.8 个跟查询案件的案由是一样的，这个结果一方面说明匹配算法能够比较好地把不同案由的案件区分开，另一方面也验证了三维评分对案件特征的刻画是有实际区分能力的。而单一的 Jaccard 方法在这个 K 值下的精确率只有 0.73。随着 K 值慢慢变大，两种方法的精确率都在往下走，原因在于测试集本身就只有 45 例案件，当 K 增加到 20 的时候，返回的结果已经占了测试集总量的百分之四十四，那么里面必然会混进更多不那么相似的同案件。在 K 等于 5 的时候，本文方法的精确率仍然保持在 0.80 这个水平上，能够满足实际应用里面那种“推荐五个类案供参考”的需求场景。

6.4 系统测试

6.4.1 白盒单元测试

白盒单元测试针对四个核心算法模块，共设计 31 条测试用例。测试人员基于代码内部逻辑结构，采用等价类划分与边界值分析相结合的方法设计用例，覆盖正常输入、边界输入和异常输入三种场景。

中文金额解析函数`parse_cn_number`是规则抽取器的关键工具函数，其测试覆盖了纯阿拉伯数字、单一中文数字、低层单位组合、含高层单位的组合以及边界与异常输入五组等价类，共 10 条用例。BIO 标签转换函数`spans_to_bio`与`get_entity_spans_from_bio`采用往返验证策略——给定 tokens 和 spans，先转换为标签序列再还原为实体，断言还原结果与原始输入一致，覆盖了单实体、多实体、空实体、末尾实体和相邻实体五种边界情况。规则抽取器的四个代表性函数测试覆盖了姓名抽取、多格式日期归一化、月工资数值提取和法条引用识别四类典型匹配场景。案件画像评分函数`generate_portrait`的测试覆盖了信息完备、信息缺失、高风险金额、违法解除、零证据和多证据六个主要条件分支。各模块的语句、分支与函数覆盖率如图 6-7 所示。

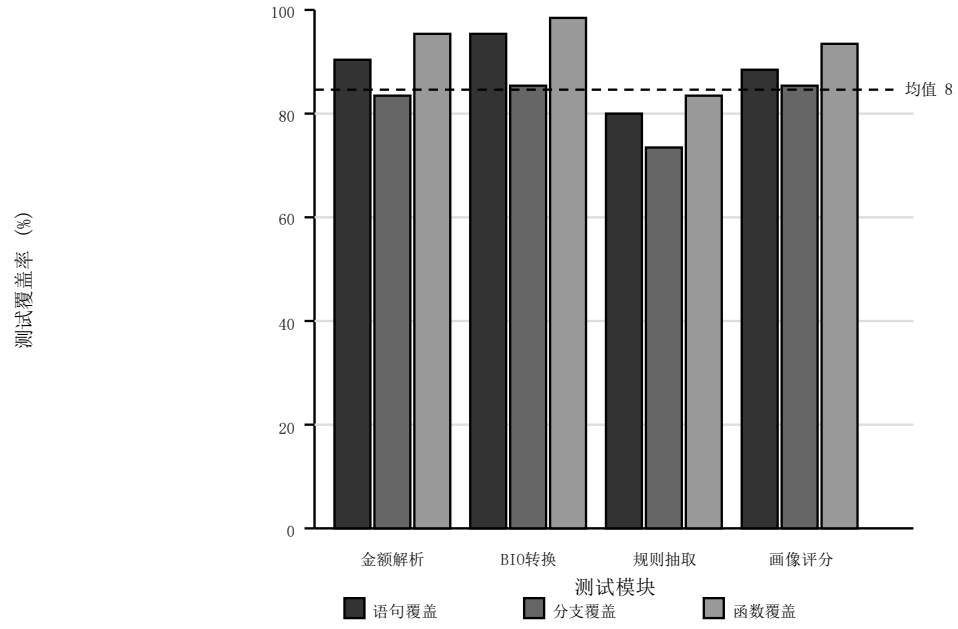

图 6-7 白盒测试覆盖率统计柱状图

31 条白盒测试用例全部通过，语句平均覆盖率 90.5%，分支平均覆盖率 83.7%，

函数覆盖率 100%。

6.4.2 黑盒功能测试

黑盒功能测试在不了解模块内部实现的前提下，通过调用系统对外提供的 API 接口验证功能正确性。API 端点测试覆盖了本系统全部 5 个核心 RESTful 端点，每个端点设计了 3 至 5 条测试用例，覆盖正常请求、边界请求和异常请求三种场景。其中，上传接口`/api/cases/upload`验证了 TXT、PDF、DOCX 三种文件格式的上传处理以及空文件的 422 响应；查询接口验证了正常查询的 200 响应和不存在资源的 404 响应。14 条 API 端点测试用例全部通过。

前端界面功能测试通过人工操作浏览器完成，验证了五个功能标签页的基本交互可用性与数据渲染正确性。上传材料页面验证了文件拖拽上传、格式校验提示和进度展示；要素抽取页面验证了要素分组表格的完整展示；案件画像页面验证了三维评分卡片、标签云和要素关系图的正确渲染；相似案件页面验证了柱状图和详情展开功能；系统评估页面验证了评估指标表格的数据展示。7 条前端界面测试用例全部通过。

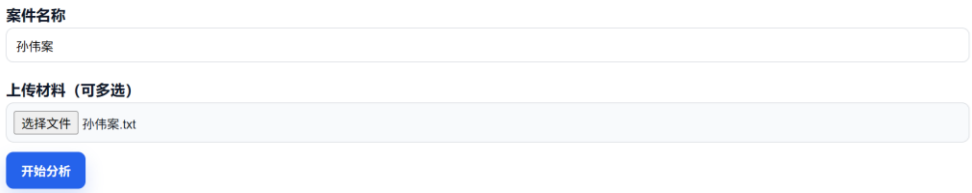

图 6-8 上传材料页面

图 6-9 要素抽取结果页面

图 6-10 案件画像页面

图 6-11 相似案件页面（1）

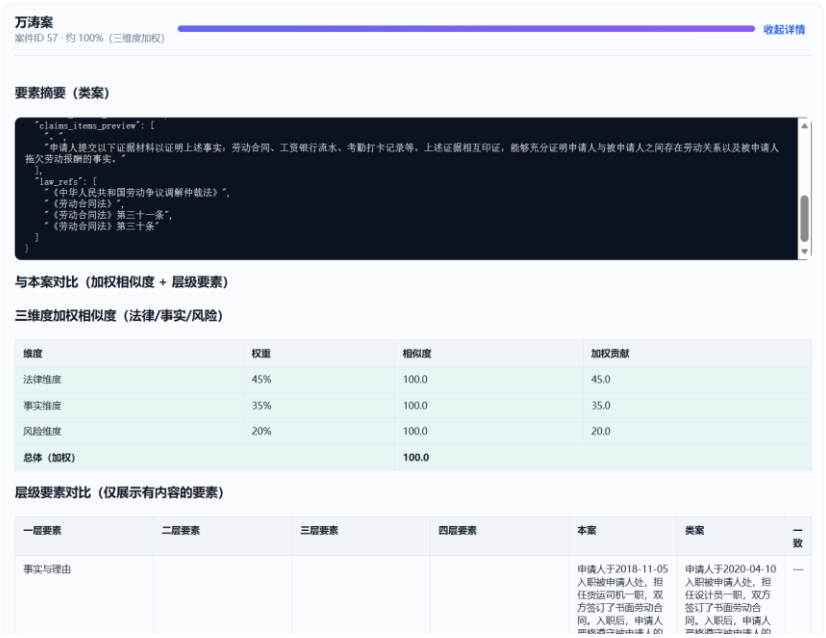

图 6-12 相似案件页面（2）

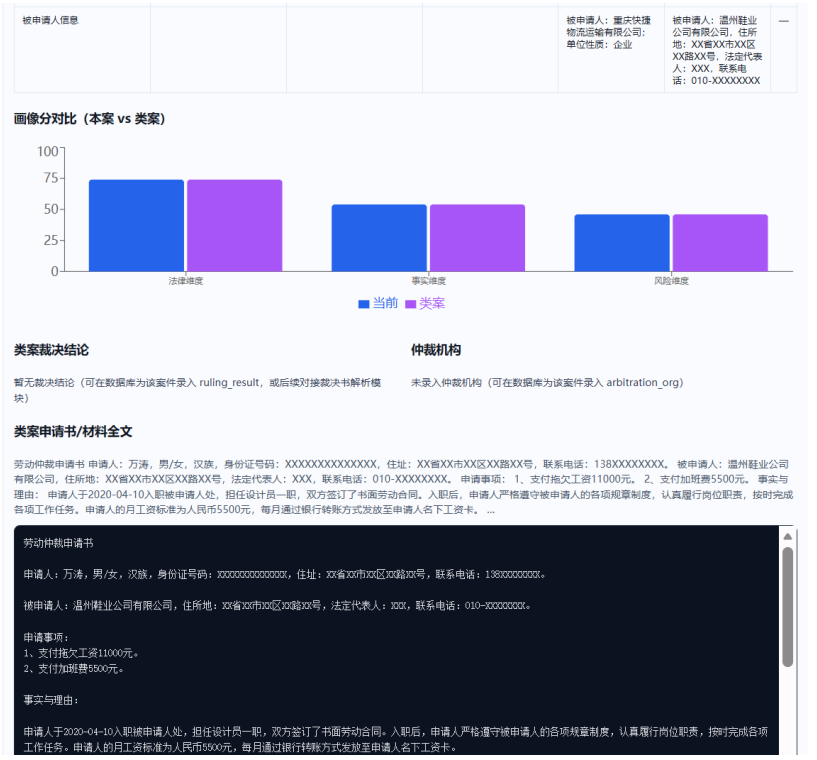

图 6-13 相似案件页面（3）

一、要素抽取指标（Precision / Recall / F1）

字段	Precision	Recall	F1
applicant_name	0.9	0.9	0.9
case_cause	0.9	0.9	0.9
claims.amount_total	1	0.1667	0.2857
claims.items	0.3333	0.125	0.1818
employer_nature	1	1	1
entry_date	1	0.7143	0.8333
law_refs	1	1	1
leave_date	1	0.7143	0.8333
month_salary	0.6667	0.6667	0.6667
overtime_desc	0.25	0.25	0.25
respondent_name	0.9	0.9	0.9
termination_reason	0.5	0.5	0.5
worker_position	0.8	0.8	0.8
_micro_avg	**0.8571**	**0.7253**	**0.7857**

二、画像构建完整性与有效性

指标	覆盖数	覆盖率
历史案件总数	34	—
portrait 已生成	32	94.1%
scores 已生成（legal/fact/risk）	32	94.1%
subscores 已生成（维度细分）	32	94.1%
keywords 已生成（词云）	23	67.7%
tags 已生成（多级标签）	29	85.3%
risk_assessment 已生成	32	94.1%
complexity_level 已生成	32	94.1%

三、相似度权重（当前配置）

维度	权重
法律维度	0.45
事实维度	0.35
风险维度	0.2

图 6-14 系统评估页面

6.4.3 端到端流程测试

端到端测试模拟了用户从案件材料上传到画像查看的完整业务链路，验证“上传材料→要素抽取→画像生成→相似案件匹配”五个步骤的串联正确性与数据传递一致性。测试使用一份涵盖完整信息的模拟劳动仲裁案件文本，依次调用后端 API 完成全流程操作，对每一步的响应状态码和返回数据的关键字段进行断言校验。端到端测试的全部 5 个步骤均顺利通过，验证了系统各模块之间的协作正确性。

结　论

本文以劳动仲裁案件为对象，围绕“材料解析—要素抽取—画像构建—系统实现”这一技术链路，完成了从领域分析到系统开发的完整研究闭环。主要工作如下。

设计了“规则+ChineseRoBERTa 多任务网络+Ollama Qwen2.5 LLM”三层混合要素抽取模型，按字段类型路由融合。实验表明，混合策略的 Micro-F1 达到 0.864，案由分类准确率为 0.89，均优于单一方法。

构建了法律、事实、风险三个维度的案件画像体系，每个维度给出 0-100 的量化评分并划分风险等级，同时整合了关键词云与多级法律标签分类，为仲裁员提供多层次的案件认知视图。

基于 FastAPI 和 React 开发了集成化 Web 原型系统，包含数据接入、要素抽取、画像构建、相似匹配和可视化展示五个模块，52 条测试用例全部通过。

本文的创新点包括：提出按字段类型路由的三层混合要素抽取架构，验证了混合策略在该领域中的有效性；构建了三维度、共 8 项子指标的量化画像评分体系，为类案匹配提供了可计算的语义基础；打通了从文档解析到画像可视化的完整技术流程。

本文也存在以下不足：训练数据仅有 150 条增强样本，NER 召回仍依赖规则层；BERT 在处理嵌套实体和边界模糊实体时稳定性不足；Ollama 零样本推理耗时较高，输出格式偶有波动；系统尚未加入裁决结果预测等深层辅助决策功能。后续工作将围绕扩充标注数据、增强 NER 边界识别、优化 LLM 稳定性以及深化系统功能等方向继续推进。

致　谢

首先，我要衷心感谢我的指导老师。在选题阶段，老师帮助我明确了劳动仲裁案件智能化处理这一兼具现实意义与技术挑战的研究方向；在课题推进过程中，老师定期跟进进展，对系统架构设计、论文结构安排等关键环节提出了大量中肯的建议；在论文撰写阶段，老师耐心审阅初稿，从行文逻辑到图表规范逐一指出问题。老师严谨的治学态度和丰富的工程经验使我受益匪浅。

感谢实验室的各位同学，在模型训练、前端调试等环节中给予的技术讨论与协助，和你们的交流帮助我解决了诸多具体问题。感谢我的室友和朋友们，在紧张的毕业季里相互鼓励、共同进退。

感谢我的家人，多年来对我学业的无条件支持与理解，你们的信任与付出是我不断前行的最大动力。

最后，感谢评阅本文和参加答辩的各位老师，在百忙之中抽出时间审阅论文并提出宝贵意见。

参 考 文 献

[1] 中华人民共和国国民经济和社会发展第十五个五年规划纲要 第四十一章 第三节

[2] 黄建洪等：苏州劳动人事争议仲裁队伍健康发展面临的问题与对策建议 黄建洪 陈伟光 2023-12-28

[3] Dozier, C., & Haschart, R. (2000). Automatic Extraction and Linking of Person Names In Legal Text. Proceedings of RIAO 2000, Paris, France, pp. 1305-1321.

[4] 曾兰兰, 王以松, 陈攀峰. 基于 BERT 和联合学习的裁判文书命名实体识别[J]. 计算机应用, 2022, 42(10): 3011-3017. DOI: 10.11772/j.issn.1001-9081.2021091565.

[5] Dong C H, Zhang J J, Zong C Q, et al. Character-Based LSTM-CRF with Radical-Level Features for Chinese Named Entity Recognition[C]//Proceedings of ICCPOL/CCF NLP-CDC 2016, LNCS 10102. Springer, 2016: 239-250.

[6] 宿亚杰. 基于领域本体的案件相似度分析方法研究[D]. 中国人民公安大学, 2022.

[7] Jiang Z., Liu B., Qian X., et al. Dynamic lexicon integration and update in BERT for enhanced named entity recognition in Chinese legal contracts[J]. International Journal of Machine Learning and Cybernetics, 2026. DOI: 10.1007/s13042-025-02961-x

[8] Geng S, Lebret R, Aberer K. Parameter-efficient legal domain adaptation[C]//Proceedings of the Natural Legal Language Processing Workshop, 2022.

[9] Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning[C]//Proceedings of the 25th international conference on Machine learning. ACM, 2008: 160-167. DOI: 10.1145/1390156.1390177

[10] Yang B, Luo X. Recent progress on named entity recognition based on pre-trained language models[C]//2023 IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, 2023: 799-804.

[11] Xiao C, Hu X, Liu Z, et al. Lawformer: A pre-trained language model for Chinese legal long documents[J].AI Open, 2021, 2: 79-84

[12] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., & Chen, W. (2022). *LoRA: Low-Rank Adaptation of Large Language Models*. In International Conference on Learning Representations (ICLR)

[13] Houlsby N, Giurgiu A, Jastrzebski S, et al. Parameter-efficient transfer learning for NLP[C]//International Conference on Machine Learning (ICML). PMLR, 2019: 2790-2799.

[14] Li X L, Liang P. Prefix-Tuning: Optimizing Continuous Prompts for Generation[J]. arXiv preprint, 2021. arXiv:2101.00190 [cs.CL]

[15] Zhao P, Han C, Han K. LawLLM-DS: A Two-Stage LoRA Framework for Multi-Label Legal Judgment Prediction with Structured Label Dependencies[J]. Symmetry, 2026, 18(1): 150. DOI: 10.3390/sym18010150

[16] 以智慧法院建设推进审判体系和审判能力现代化——最高法举行人民法院智慧法院建设工作成效新闻发布会 来源：人民法院报 发布时间：2022-10-14

[17] 肖朝霞, 王波, 李强. 面向民事案件的画像框架构建研究[J]. 计算机工程与应用, 2024

[18] 陈宝贵, 人民法院信息技术服务中心. 全案件类型知识图谱库[DS/OL]. 国家基础学科公共科学数据中心, 2024. CSTR:16666.11.nbsdc.pB6Qcy5D

[19] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-Rank Adaptation of Large Language Models. International Conference on Learning Representations.

[20] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. arXiv Preprint arXiv:2305.14314.

附录 A 译文

AGB-DE：德国消费者合同条款自动法律评估语料库

丹尼尔·布劳恩　特温特大学高科技商业和创业系 d.braun@utwente.nl
弗洛里安·马蒂斯　慕尼黑理工大学计算、信息和技术学院 matthes@tum.d

摘要

法律任务和数据集经常被用作语言模型能力的基准。然而，公开可用的带注释的数据集很少。在本文中，我们介绍了 AGB-DE，一个由 3764 条条款组成的语料库，这些条款来自德国消费者合同，已经由法律专家进行了注释和法律评估。结合这些数据，我们提出了检测潜在无效子句任务的第一个基线，比较了支持向量机基线与三个微调的开放语言模型的性能和 GPT-3.5 的性能。我们的结果显示了这项任务的挑战性，没有一种方法超过 0.54 的 F1 分数。虽然微调的模型在精确度方面往往表现得更好，但 GPT-3.5 在召回率方面优于其他方法。对错误的分析表明，主要挑战之一可能是正确解释复杂的条款，而不是决定什么是允许的，什么是不允许的。

1 简介

标准格式消费者合同，即由公司单方面起草的消费者合同，对我们的经济具有巨大的意义，也对消费者保护具有重大意义。他们的审查是一项艰巨的任务，由公司、律师事务所、政府组织和非政府组织(NGO)执行。近年来，研究人员研究了不同的计算方法来自动化部分合同审查过程(见第 2 节)。

此类研究面临的一大挑战是缺乏合同数据，尤其是带注释的数据。对于英语以外的其他语言来说更是如此。用法律评估对合同进行注释是一项费力的过程，只能由高资质的专家执行，因此费用非常昂贵。大语言模型(LLM)的商业提供商，如 OpenAI，以及科学界，越来越多地使用法律任务来评估(或促进)LLM 的能力。对于许多现有的数据集和任务，很难怀疑像 GPT3.5 这样的 LLM 没有接受过培训，这将质疑对它们进行的评价的有效性(BalLoccu 等人，2024 年)。

在本文中，我们提出了一个新的语料库，包括 3,764 个子句，11,387 个句子，以及 250,859 个来自德国消费者合同的记号。法律专家对每一条款都作了注释，注明了条款主题以及条款是有效的还是可能无效的(见第 3.2 节)。合同条款是合同中通常通过格式化与其他条款明确分开并涉及特定条款的一节。该语料库总共包含 8,582 个标签，可在 GitHub 和 Hugging Face 数据集上使用。

此外，我们还给出了识别潜在无效子句的任务的基线，比较了支持向量机、三个不同大小的开放语言模型和 GPT-3.5。我们的结果显示了这项任务的挑战性，没有一种方法超过 0.54 的 F1 分数。性能最好的型号 AGBert 也可供下载。虽然微调的模型在精确度方面往往表现得更好，但 GPT-

3.5 在召回率方面优于其他方法。对错误的分析表明，主要挑战之一可能是正确解释复杂的条款，而不是决定什么是允许的，什么是不允许的。用于准备数据集和训练模型的代码也可以在 GitHub 上找到。我们希望语料库将有助于实现未来开放的和可重复的自然语言处理研究。

2 相关工作

在过去的几年里，法律任务和数据集对语言模型的评估变得越来越重要。从特定领域的模型，如 Legal-Bert(Chalystis 等人，2020 年)，到基准数据集，如 LexGLUE(Chalystis 等人，2022 年)，最终在 OpenAI 提交 GPT-4 时达到顶峰，其中所谓的模型的“法律技能”一直是表明其比前一版本有所改进的主要传播点之一(Achiam 等人，2023 年)。此外，GPT 模式是否能够通过法学院(Choi 等人，2021 年)或通过律师考试(Katz 等人，2023 年)的问题在过去几年里一再被提出。

2.1 合法数据集

一般来说，有大量的合法数据集可用。主要是因为许多政府和机构以数字方式和在开放许可下发布法律、法院裁决和类似的法律文件。然而，其中只有一小部分被手动注释(Braun，2023)。因此，法律领域中的许多受监督的 NLP 任务依赖于固有地包含标签或标签可以自动派生的语料库。例如，在可以使用欧洲联盟平行语料库的情况下进行翻译(Skadin，š 等人，2014 年)或对法庭案件进行结果预测，在这种情况下，可以很容易地从判决文件中提取最终判决(Chalystis 等人，2019 年)。而在语料库中很少找到政府或机构不定期发布的文档类型，如(消费者)合同。

由法律专家手动注释非常昂贵，因此很少见。虽然存在一些这样的数据集(见第 2.1.1 和 2.1.2 节)，特别是在英语中，但我们不知道有哪些不同语言的语料库的规模可以与 AGB-DE 语料库相媲美，后者由 93 份合同中的 3,764 条法律注释条款组成。

2.1.1 消费者合同

由 Ruggeri 等人提供的语料库包含 100 个英文服务条款和总共 1,715 个带注释的从句(2022)是同类项目中规模最大的之一。条款分为公平条款和不公平条款，在不公平条款中分为五大类。然而，Drawzeski 等人(2021)提供了一个类似规模的 100 个服务条款语料库，其中只有 25 个不同的合同，每一个合同都有四种语言(英语、意大利语、德语和波兰语)。

另一个大型消费者合同语料库是威尔逊等人的 OP-115 语料库(2016)，其中包括 115 项来自网站的隐私政策，这些网站已由法律系学生用文本中出现的数据实践进行注释。Mapp 语料库是一个更大的隐私策略语料库，包含来自移动应用程序的 155 条隐私策略(Arora 等人，2022 年)，这些策略已经以类似的方式进行了注释。这两种都只包含英文合同。

2.1.2 B2B 合同

在商业(企业对企业)合同领域，可以使用更大的数据集。Hendrycks 等人的 CUAD 数据集(2021)由 510 份英文商业合同组成，其中 41 种不同类型的法律条款已作了注释。查尔基迪斯等人(2017)提供一个数据集，其中包括 993 份标有条款标题的合同和 2461 份由法律系学生标注了其他类型

合同要素的合同。与 Ruggeri 等人的数据集(2022)以及 AGB-DE 语料库不同，这些数据集没有注明法律判决，而是关于个别条款的主题。

2.2 合同条款的法律评估

上述的一些数据集中，还有其他的，主要是未公布的数据集，已经被用来训练不同的 NLP 模型，以便预测给定合同条款是否有效。这也是我们在建立 AGB-DE 语料库时主要考虑的任务。Ruggeri 等人(2022 年)使用记忆增强神经网络来检测服务方面的不公平条款，并报告了 0.526 的准确率。Braun 和 Matths(2021)使用微调的 Bert 模型预测在线商店条款和条件中的无效条款，报告的准确率为 0.9%。托瑞等人。(2020)使用支持向量机检测隐私策略中缺失的条款，报告的精确度为 0.85，召回率为 0.96。最近，马丁等人(2024)提出了一项研究，除其他外，比较了 GPT4-32k 与初级律师在寻找合同中的法律问题方面的表现。他们报告说，GPT4(F1-得分 0.74)不仅更快、更便宜，而且比初级律师(F1-得分 0.667)更好。

3 语料库

语料库由 93 份标准格式消费者合同中的 3,764 条条款组成。这一部分描述了语料库的构建。语料库的详细数据表(Gebru 等人，2021 年)可在附录 F 中找到。

3.1 数据

这些数据是在 2021 年和 2022 年收集的，并在 2021 年和 2023 年之间进行了注释。它包括可在线获得的德国标准格式消费者合同，如在线商店、健身工作室和电信提供商的条款和条件。这些数据是由注释者、消费者保护律师(更多细节见第 3.2 节)根据他们的专业兴趣收集的。每个合同中的每个条款都被手动复制到一个 Excel 文件中，以及条款标题和合同文本的 URL。通过这种方式，总共收集了 93 份合同。

3.2 注释

该数据集由来自两个德国非政府组织的五名完全合格的律师注释，重点是消费者保护。每个解说员都有多年的消费者保护法和消费者咨询经验。批注是在 Excel 文件中进行的，与专用批注工具相比，批注者更喜欢使用 Excel 文件。

3.2.1 主题

首先，为每个子句添加一个或多个主题标签。随后，可以添加副主题标签以进一步指定子句的内容。对于这种分类，我们使用了 Braun 和 Matths(2022)介绍的分类方法，它由 23 个主题标签和 37 个副主题标签组成(可用标签的列表见附录 A)。虽然每个子句必须至少用一个主题标签进行注释，但注释者被指示只在他们认为合适的地方添加副主题。

3.2.2 有效性

之后，每一条条款都由专家注释员进行法律评估。合同中的条款如果与管辖法律相抵触，则无效，即合同当事人不能强制执行。一项条款是否实际无效取决于许多因素，包括在某些情况下，

一方当事人是否为消费者或双方是否都是企业。关于特定情况下的特定条款实际上是否是无效的最终决定只能由法院做出。因此，对注释者的指示是，如果他们认为居住在德国的消费者可以在法庭上成功挑战某一条款，则将该条款标记为可能无效。对于本文的其余部分，如果我们说一个子句是无效的，我们的意思是它被专家注释者注释为可能无效。除了评估本身，它是二进制的(1-可能无效，0-有效)，注释者可以在 Excel 文件中添加注释来解释他们的决定。根据注释者的意愿，这些解释不在出版的语料库中。值得再次强调的是，这些注释员为致力于倡导消费者权利的非政府组织工作，因此他们对法律的解释可能比为大公司工作的律师更有利于消费者。

最初，两名专家(每个组织一名)对数据的一小部分进行了注释。在这个最初的注解中，76%的案例法律注解是一致的。在这些初始注释的基础上，专家们讨论并调整了他们的注释策略，以增加注释的一致性。在这个过程中，分歧背后的两种模式变得明显起来。在这一阶段发现的所有分歧都是基于对法律或法院裁决的解释的分歧，而不是对条款的解释的分歧，即合同的文本。第二种模式是，许多分歧是基于使用模糊法律术语的法律。法律往往是故意含糊其辞地制定的。例如，《德国民法典》认为规定“不合理长的付款期限”的条款无效(第 308 条，第 1A 条)。看似糟糕的法律制定是为了使法律成为“未来的证据”。例如，付款截止日期是否过长，在 20 世纪 70 年代和今天发生了重大变化，当时信件和银行转账仍然需要数天时间。这些术语由法院来解释，这些解释也可能随着时间的推移而改变。为了使注释者更加一致，对注解准则进行了扩展，增加了对这种模糊法律术语的相关法律条款的商定解释。此外，作为一种“包罗万象”的解决方案，还引入了一项规则，即当有疑问时，例如因为不同的法院做出不同的裁决，注释者将总是使用对消费者更友好的解释。对于随后的完整语料库的标注，每个实例都由一个注释器进行标注。虽然每个实例有多个注释器会更好，但由于成本原因，这是不可行的。对农业标准数据库语料库进行数据收集和注释的总费用约为 110.000 欧元。

3.3 匿名化

出于道德和法律方面的原因，我们决定在发布数据集之前将其匿名。虽然原始数据不包含来自个人的任何信息，但它确实包含来自公司的(可公开访问的)信息，如电话号码、地址和税务 ID。我们采取了多个步骤从合同中删除这些数据，以便更难识别起草合同的公司。随着时间的推移，公司可以而且确实会改变他们的合同，所以我们希望避免消费者发现并阅读过时版本的合同。此外，注释者所作的一项或多项评估不可能在法庭上站得住脚，要么是由于无意识的错误，要么是由于上述关于法律解释的偏见。错误地声称一家公司使用无效条款可能会对他们的业务造成损害，并可能牵涉到责任。另一方面，如果消费者依赖这种评估，错误地声称一项条款是有效的可能会损害他们的利益。因此，我们实施了十个匿名化步骤：

1.从语料库中删除主题标签为 PARTY 的所有子句(这些子句仅包含关于签约方的信息，即公司)

2.使用正则表达式(Regex)将所有电子邮件地址替换为“Hello@Example.com”

3.使用正则表达式将所有 URL 替换为“www.Example.com”

4.使用正则表达式将所有国际银行帐号(IBANS)替换为“DE75512108001245126199”

5.使用正则表达式将所有税务 ID 替换为 DE398517849

6.使用正则表达式将所有电话号码替换为 00 00 12345678

7.使用正则表达式将所有邮政编码替换为 00000

8.使用命名实体识别(NER)将所有公司和组织的名称替换为“«name”

9.使用 NER 将所有城市名称替换为“«Stadt»”(德语中城市的意思)

10.使用 NER 将所有街道名称替换为“«Strasse»”(德语中街道的意思)

虽然前七个步骤被证明是非常有效和直接的(总共删除了 84 个政党条款，并替换了 120 个电子邮件地址、231 个 URL、2 个 IBAN、117 个电话号码和 279 个邮政编码)，但许多可用的标准 NER 库被证明对文本不是很有效。最终，FLAIR 图书馆(Akbik 等人，2019 年)与 NER-德国法律模式(Leitner 等人，2019 年)被证明是最合适的。在该模型的帮助下，我们能够替换 724 个组织和公司的名称，418 个城市名称和 53 个街道名称。然而，人工检查显示，另一轮手动匿名是必要的。在此手动过程中，还删除了另外 1338 个公司名称、38 个城市名称和 85 条街道。

为了避免可能导致分类性能下降的过度匿名化，组织和 URL 列表被明确排除在删除或替换之外。这份名单主要包括欧盟等政治机构及其 URL、航运公司及其 URL、以及支付提供商及其 URL。表 1 显示了最终语料库的摘录。

4 语料库分析

语料库包括 93 个合同，有 3,764 个条款(平均每份合同 40 个条款)，包含 11,387 个句子(平均每个子句 3 个句子)和 250,859 个令牌(平均每句 22 个)。在语料库中出现的 3764 个子句中，179 个(4.8%)被注释为潜在无效。这是可比的，尽管略低于布劳恩和马特在一个小得多的 24 份合约的数据集上报告的 6%(2021 年)。虽然这导致了一个不平衡的语料库，但我们认为这是对现实的现实反映，在现实中，无效从句的出现频率也明显低于无效从句。

表 2 显示了子句在主题中的分布情况以及每个主题中可能无效的子句所占的百分比。由于一个子句可以属于多个主题和子主题，因此标签的总和大于子句的数量。30 个从句都有不止一个主题的注解。在这 30 个子句中，一个子句被标注了三个主题，一个子句被标注了四个主题标签，另外 28 个子句被标注了两个主题标签。在 3764 个子句中，只有 1078 个子句有副主题注释。在某种程度上，我们认为这是因为注释者不一定要增加一个小主题，而且注解的重点显然是法律评估。语料库总共包含 8,582 个标签。

对一主题有关的无效从句的分布情况的分析表明，某些类别特别容易被专家视为潜在无效。通过与专家进一步深入分析发现，这些条款往往属于受到严格规制的类型。例如，关于类别变更

的条款，涵盖了合同生效后对其所做出的各项修改。鉴于格式合同已被认为"反映了缔约双方权力的不平衡"（Braun 等人，2019），企业在客户同意后单方面修改合同的可能性受到严格规制也就不足为奇。因此，标注者频繁认定此类条款无效是完全合理的。类似的逻辑也适用于其他主题，如责任限制条款或合同可分割性条款等。

Id	Con.	Lang	Title	Text	Topics	Subtopics	Void
10	1	de	2. Widerrufs-belehrung	Sie tragen keine Kosten für die Rücksendung der Ware.	withdrawal	withdrawal: shipping-Costs	0
124	3	de	11. Gel-tungsbedin-gungen der AGB	Anstelle der unwirk-samen Vorschrift gilt eine Regelung, die der mit der unwirksamen Vorschrift verfolgten wirtschaftlichen Zweck-setzung am nächsten kommt.	severability		1
127	4	de	1. Allge-meines	1.4. Mit der Bestellung auf dieser Website bestätigen Sie, dass Sie volljährig und rechtsfähig sind, einen Verbrauchervertrag abzuschließen."	age		0
215	9	de	§6 - EIGEN-TUMSVOR-BEHALT	Die von «NAME» gelieferte Ware verbleibt bis zur vollständigen Bezahlung Eigentum von «NAME»	ret.OfTitle		0

Table 1: Excerpt from the corpus

5 自动法律评估

为了给出语料库设计的主要任务的基线，并评估任务的难度，我们比较了不同的语言模型和支持向量机将子句分为(潜在的)无效和有效。

5.1 数据拆分

为此，我们从语料库创建了一个数据集，该数据集被分为 80%(3004 个子句)的训练集和 20%(755 个子句)的测试集。为确保两个数据集中各分类的代表性均衡，我们依据主题标签和法律评估结果对数据进行了分层划分。原始语料库中有五条条款在此数据集中被移除，原因是这些条款在语料库中属于其条款类型的唯一无效实例，因此无法按照上述方式进行划分。

Label	Amount	Void (%)
age	20	0.00
applicability	148	2.03
applicableLaw	87	3.45
arbitration	97	1.03
changes	9	11.11
codeOfConduct	29	0.00
conclusionOfContract (cOc)	557	5.92
cOc:binding	50	0.00
cOc:changeOfOrder	1	0.00
cOc:definition	39	0.00
cOc:restrictions	13	0.00
cOc:steps	59	0.00
cOc:withdrawal	17	0.00
contractLanguage	41	0.00
delivery	475	7.16
delivery:brokenPackaging	19	0.00
delivery:costs	53	0.00
delivery:customs	2	0.00
delivery:destination	18	0.00
delivery:methods	26	0.00
delivery:partial	22	0.00
delivery:time	1	0.00
description	46	0.00
disposal	36	0.00
intellectualProperty	39	0.00
language	9	11.11
liability	211	9.00
party	0	0.00
payment	642	6.07
payment:fee	6	0.00
payment:late	27	0.00
payment:loyalty	1	0.00
payment:methods	289	0.00
payment:restraint	5	0.00
payment:vouchers	124	0.00
personalData	115	0.87
personalData:cookies	6	0.00
personalData:duration	0	0.00
personalData:information	1	0.00
personalData:reason	1	0.00
personalData:update	0	0.00
personalData:usage	1	0.00
placeOfJurisdiction	53	1.89
prices	147	1.36
prices:currency	6	0.00
prices:vat	36	0.00
retentionOfTitle	125	2.40
severability	35	11.43
textStorage	57	1.75
warranty	314	6.37
warranty:options	4	0.00
warranty:period	10	0.00
withdrawal	506	3.75
withdrawal:compensation	7	0.00
withdrawal:effects	45	0.00
withdrawal:exclusion	51	0.00
withdrawal:form	25	0.00
withdrawal:model	4	0.00
withdrawal:period	33	0.00
withdrawal:shippingCosts	11	0.00
withdrawal:shippingMethod	7	0.00
Total lvl 1	3798	4.80
Total lvl 2	1020	

Table 2: Distribution of topics and void clauses

5.2 抽样不足

由于我们清楚地意识到该数据集因其不平衡特性将带来挑战，因此我们创建了第二个数据集。在该数据集中，我们采用欠采样技术（Liu 等人，2008）来削减过度代表性类别的数据。具体而

言，每个主题与有效性组合的实例数量被限制在 100 条以内。例如，若某主题包含 40 条无效条款和 120 条有效条款，则仅有 100 条有效条款会被纳入这第二个数据集。通过这种方式，我们最终得到一个包含 1,362 条（80%）训练条款和 345 条（20%）测试条款的数据集。数据划分方式保持不变——即我们仅移除数据，但未改变训练集与测试集之间已有数据的分布结构。为便于区分，我们将第一个较大的数据集称为 agb-de，而将欠采样数据集称为 agb-de-under。

5.3 模型

我们利用这两个数据集训练了一个支持向量机模型，该模型以条款的 TF-IDF 向量作为输入；同时我们还对以下四个不同的语言模型进行了微调与评估：

- BERT 模型 bert-base-german-cased（Chan 等人，2020），该模型此前已在包含超过 10 万份德国法律文档的"开放法律数据"数据集上进行了训练（Ostendorff 等人，2020）
- 多语言 RoBERTa 模型 xlm-roberta-base（Conneau 等人，2019）
- 德语 GPT2 模型 gerpt2（Minixhofer，2021）
- 以及 gpt-3.5-turbo-0125，该模型以"零样本"方式进行评估，未进行微调

针对三个开源模型的微调，我们进行了人工超参数搜索，初始值采用各模型的标准超参数配置。最终微调所用的超参数设置详见附录 B，同时也发布在随语料库公开的训练代码中。为应对数据集不平衡问题，我们还采用了定制化的损失函数。在所有模型的微调过程中，损失函数中有效条款的类别权重设为 1.0，无效条款的类别权重设为 100.0。用于评估 GPT-3.5 的提示语与 API 调用方式详见附录 C。

6 评估

评估结果如表 3 所示。我们重点关注的核心评估指标是 F1 分数，该指标能在两个分布不均的类别间取得平衡。

针对 agb-de 数据集，所有模型均未能有效处理其高度不平衡的数据特性。在该数据集上表现最佳的模型是经过微调的 BERT 模型。虽然 RoBERTa 模型通常在处理不平衡数据方面优于 BERT（Younes 与 Mathiak，2022），但我们所微调的 RoBERTa 是多语言模型，而微调的 BERT 模型则专门在德语法律数据上进行过预训练。表现第二佳的是支持向量机模型，其 F1 分数为 0.31。对于 GPT-3.5 而言，其性能表现与数据集差异关联不大：在两个数据集上，GPT-3.5 均实现了最高的召回率，但代价是精确率极低——即 GPT-3.5 将大部分条款错误地分类为可能无效（另见图 2d）。

所有模型在 agb-de-under 数据集上的表现均有提升，这表明欠采样是该语料库适用的处理策略。其中 BERT 模型再次表现最佳，取得了 0.54 的 F1 分数。为验证欠采样数据集带来的性能提升究竟是源于模型优化，还是仅仅因为测试集中偶然剔除了困难样本，我们还使用**agb-de**数据集的原始测试集对所有基于欠采样数据集训练的模型进行了测试。该评估结果详见附录 D。除支持向量机外，所有模型在欠采样数据集上训练后，于完整数据集上评估的表现均优于直接在完

整数据集上训练和测试的结果。这表明，通过移除过度不平衡的训练数据，确实能在此数据集上构建出更优的模型。

7 误差分析

图 1 展示了各模型在欠采样数据集上对语料库中主要主题类别的精确率与召回率。数据显示，不同模型在各个主题上的表现差异显著。以"责任限制"条款为例，所有模型的召回率均未超过 0.25——该主题中无效条款占比达 11%，是主要主题中无效比例最高的类别。对专家标注者而言，由于存在非常明确的法律规定（例如《德国民法典》第 309 条第 7 款明确指出"因使用方过失导致人身伤害的责任排除或限制条款无效"），责任条款的标注相对容易。然而，语言学分析显示，责任条款的表述结构往往复杂，常在一个句子中包含多重明示的涵盖与排除情形，这可能是导致大多数模型在该类别上表现欠佳的原因。

虽然同一主题的条款可能因不同原因而无效，但对于某些主题而言，存在一些主要的无效模式。例如，在本语料库中，质保条款主要因限制用户在过短的指定时限内报告缺陷而无效。类似地，付款条款主要因对逾期付款设置过高费用而无效。相较于其他更具细微差别的主题，BERT 模型能更有效地识别这些重复性较强的模式。GPT-2 模型在付款条款这一语料库中占比最大的类别上取得了最高精确率，因此显著影响了整体结果。总体而言，各类别间的性能差异表明，将主题标签作为附加特征可能有助于提升分类性能。GPT-3.5 的表现与其他模型存在差异，它更容易产生误判，即将实际有效的条款错误地标记为无效。我们用于 GPT-3.5 的提示不仅要求其将条款分类为可能有效或无效，还要求模型提供"解释"。虽然这些生成的文本并非真正意义上的解释（即未能清晰阐明评估依据），但有趣的是，其中描述的问题大多数情况下是正确的——然而无论是生成文本还是标注结果，其最终判断仍是错误的。以一条关于合同撤销权的条款（语料库 ID 5）为例，GPT-3.5 的"解释"为："该条款可能无效，因其可能对消费者造成不公平损害。根据法律规定，消费者必须能够清晰且显著地行使撤销权，不得附加任何额外障碍或条件。此条款规定通过'我的账户'功能登记退货视为行使撤销权，这可能限制消费者行使撤销权的能力。重要的是消费者应能在无额外义务的前提下行使撤销权。"（德语原文参见附录 E.1）尽管这段论述本身在事实上是正确的，却并不适用于当前条款。该条款明确描述了行使合同撤销权的其他方式，并特别声明使用"我的账户"功能并非强制要求。与文本描述一致的是，GPT-3.5 仍错误地将该条款标注为可能无效。

类似地，针对条款"若客户为商户……本合同产生的一切争议管辖法院为卖方营业所在地法院……"（语料库 ID 3212），GPT-3.5 判定该条款可能无效，理由是"该条款使消费者处于不合理劣势，因其将消费者约束至企业指定的管辖法院"。其法律推理本身再次正确，但并不适用于该条款——因为该条款仅适用于商户情形（此时条款有效）。为测试 GPT-3.5 是否知晓该条款对商户有效，我们调整了提示语，要求系统设想自己作为商户（而非消费者）的律师，其回复果然变为

"若客户为经营者，该条款不太可能无效"。多数情况下，GPT-3.5 的主要挑战似乎并非缺乏法律"知识"，而是对条款的错误解读，尤其当条款包含选择性内容或并非适用于所有客户时。

大语言模型的另一个典型问题可以在针对语料库 ID 10 条款生成的解释中观察到。模型生成的文本如下："根据《德国民法典》（BGB）第 357 条，消费者行使撤销权时不得被要求承担退货运费。因此，要求消费者承担退货运费的条款可能无效。"（参见附录 E.2）。在 2014 年之前，确实存在"消费者行使撤销权时不得被要求承担退货运费"的规定（至少针对 40 欧元以上的购物）。然而，现行《德国民法典》第 357 条已明确规定相反内容，即消费者需自行承担退货运费。由于 GPT-3.5 的训练数据截止至2021年，其中涉及旧法规的文本可能更为常见，这导致了该错误认知。大语言模型的另一个典型问题可以在针对语料库 ID 10 条款生成的解释中观察到。模型生成的文本如下："根据《德国民法典》（BGB）第 357 条，消费者行使撤销权时不得被要求承担退货运费。因此，要求消费者承担退货运费的条款可能无效。"（参见附录 E.2）。在 2014 年之前，确实存在"消费者行使撤销权时不得被要求承担退货运费"的规定（至少针对 40 欧元以上的购物）。然而，现行《德国民法典》第 357 条已明确规定相反内容，即消费者需自行承担退货运费。由于 GPT-3.5 的训练数据截止至 2021 年，其中涉及旧法规的文本可能更为常见，这导致了该错误认知。

Dataset	Model	Precision	Recall	F1-score
agb-de	svm	0.37	0.27	0.31
	bert-base-german-cased	0.50	0.27	**0.35**
	xlm-roberta-base	0.00	0.00	0.00
	gerpt2	**0.71**	0.14	0.23
	gpt-3.5-turbo-0125	0.06	**0.92**	0.11
agb-de-under	svm	0.40	0.32	0.36
	bert-base-german-cased	0.51	0.57	**0.54**
	xlm-roberta-base	**0.75**	0.08	0.15
	gerpt2	0.64	0.43	0.52
	gpt-3.5-turbo-0125	0.13	**0.92**	0.22

Table 3: Results of the evaluation, best performance on a dataset is highlighted in bold

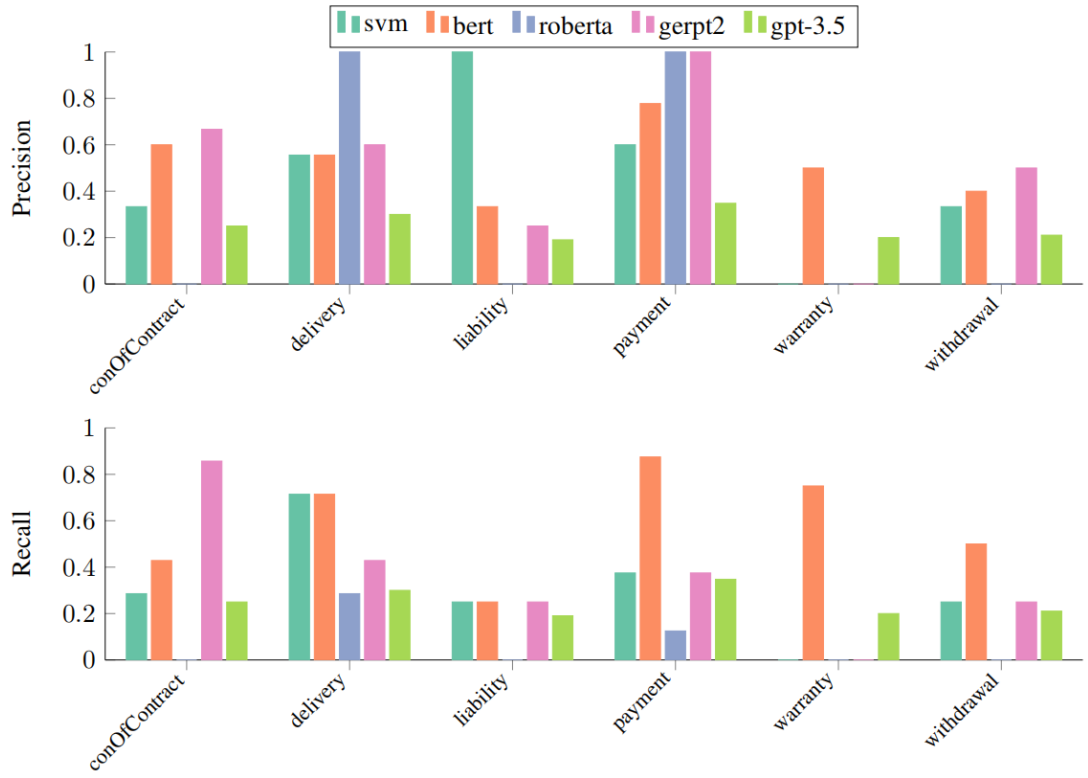

Figure 1: Precision and recall per topic on the agb-de-under dataset

Actual	Predicted: valid	Predicted: void
valid	288	20
void	16	21

(a) bert-base-german-cased

Actual	Predicted: valid	Predicted: void
valid	307	1
void	34	3

(b) xlm-roberta-base

Actual	Predicted: valid	Predicted: void
valid	299	9
void	21	16

(c) gerpt2

Actual	Predicted: valid	Predicted: void
valid	71	237
void	3	34

(d) gpt-3.5-turbo-0125

Figure 2: Confusion matrices for the evaluation on the agb-de-under dataset

8 结论

本论文介绍了 AGB-DE 语料库，该库包含由法律专家标注的 3,764 条德国消费者合同条款。除语料库外，我们还提供了两个从中衍生的数据集，均已划分为训练集和测试集。这些数据集可便捷地用于机器学习模型的训练或微调。我们利用这些数据集为条款有效性分类任务建立了基准性能评估。

针对反映语料库原始分布的数据集，研究表明语言模型难以处理不平衡数据，其 F1 分数均未超过 0.35。而在通过欠采样实现更平衡分布的第二数据集中，像 BERT 和 GPT2 这样的开源模型能更好地识别无效条款，分别取得 0.54 和 0.52 的 F1 分数。在两个数据集上，开源模型的 F1 分数均优于 GPT-3.5——后者因产生大量误判，在不平衡程度较高的数据集上仅获得 0.11 的 F1 分数，在较平衡数据集上也仅为 0.22。

行为准则

错误的法律陈述始终具有潜在危害性。尽管我们与经验丰富的专家合作，对每条标注都进行了审慎判断，但数据集仍可能存在误差。若错误判定条款无效可能对企业造成负面影响，而错误认定条款有效则可能损害消费者权益。为降低此类影响，我们采取的一项措施是对条款进行匿名化处理，以增加将其与特定企业关联的难度。许多条款（如合同可分割性条款或责任限制条款）具有高度标准化特征，往往直接源自格式合同模板。在此类情形下，删除显性标识符即可实现完全匿名化。然而在其他情况下，特别是涉及奖励积分计划的条款中，即使移除显性标识符，条款内容仍可能因过于具体而被追溯至特定企业（若该企业仍使用相同条款）。我们认为其实际影响相当有限，因为从数据集中检索特定条款信息的行为，不太可能成为大量潜在客户的实际需求。我们已在语料库发布版本的说明文件中添加解释性文字，强调标注结果基于专家个人判断而非具体司法裁决，因此不具备法律约束力。总体而言，我们认为该数据集的发布有助于改善消费者与企业之间现有的权力不对等现象。结合匿名化处理策略，这一举措所承担的虽小但确实存在的风险是合理的。

同时，若企业明知其使用可能无效且损害消费者权益的条款却未采取任何措施，这种行为可能被视为不道德的。因此，与我们合作的非政府组织在标注过程中若发现对消费者极为不利、需要采取法律行动的条款时，确实会启动法律程序。

总体而言，该数据集由公开可获取的数据构成，这些数据通常由律师精心起草。从伦理角度来看，我们在数据集中未发现任何存在问题的实例。

局限

受实际条件限制，本研究构建的数据集存在若干局限性：

• 数据集从消费者保护视角进行标注，因此很可能倾向于以有利于消费者的方式解释现行法规

• 每条数据仅由一位专家标注。法律决策本身具有不确定性与解释空间，理想情况下应由多位专家进行交叉标注

• 语料库评估基于标注时期（2021-2023 年）的法律规定，基于更新数据训练的模型可能因法规修订或新判例而产生不同预测结果

• 尽管我们认为数据集在一定程度上反映了现实分布，这也意味着其存在不平衡性且无效条款占比较小

同时，本次评估工作也存在以下局限：

• 使用 GPT-3.5 等云端模型始终存在结果可复现性风险，我们通过指定模型版本号的方式尝试降低该风险

• 将针对特定任务微调的模型与零样本提示方法进行比较可能存在不对等性。未来提升 GPT-3.5 性能的策略可包括采用少样本学习或提示词工程。鉴于其错误类型，少样本方法效果可能有限，但我们相信提示词工程或能有效降低大量误判情况。

附录 B 外文原文

AGB-DE: A Corpus for the Automated Legal Assessment of Clauses in German Consumer Contracts

Daniel Braun
University of Twente Department of High-tech Business and Entrepreneurship
d.braun@utwente.nl
Florian Matthes
Technical University of Munich TUM School of Computation, Information and Technology
matthes@tum.de

Abstract

Legal tasks and datasets are often used as benchmarks for the capabilities of language models. However, openly available annotated datasets are rare. In this paper, we introduce AGB-DE, a corpus of 3,764 clauses from German consumer contracts that have been annotated and legally assessed by legal experts. Together with the data, we present a first baseline for the task of detecting potentially void clauses, comparing the performance of an SVM baseline with three fine-tuned open language models and the performance of GPT-3.5. Our results show the challenging nature of the task, with no approach exceeding an F1-score of 0.54. While the fine-tuned models often performed better with regard to precision, GPT-3.5 outperformed the other approaches with regard to recall. An analysis of the errors indicates that one of the main challenges could be the correct interpretation of complex clauses, rather than the decision boundaries of what is permissible and what is not.

1 Introduction

Standard form consumer contracts, i.e. consumer contracts that are drafted unilaterally by a company, have huge significance for our economy, but also for consumer protection. Their review is a laborious task, performed by companies, law firms, governmental organizations, and nongovernmental organizations (NGOs). In recent years, researchers have investigated different computational approaches to automate parts of the contract reviewing process (see Section 2).

One of the big challenges for such research is the scarcity of contract data in general and annotated data in particular. Even more so for languages other than English. Annotating contracts with legal assessments is a laborious process that can only be performed by highly qualified experts and is therefore very expensive. Commercial providers of Large Language Models (LLMs), like OpenAI, but also the scientific community, increasingly use legal tasks to assess (or promote) the capabilities of LLMs. For many of the existing datasets and tasks, it is questionable whether LLMs like GPT3.5 have not been trained on them, which would question the validity of evaluations performed on them (Balloccu et al., 2024).

In this paper, we present a new corpus consisting of 3,764 clauses, 11,387 sentences, and 250,859 tokens from German consumer contracts. Each clause has been annotated by legal experts with a clause topic and whether the clause is valid or potentially void (see Section 3.2). A contract clause is a section in

a contract that is usually separated explicitly through formatting from other clauses and deals with a specific provision. The corpus contains a total of 8,582 labels and is available on GitHub1 and as Hugging Face dataset 2.

In addition, we present a baseline for the task of identifying potentially void clauses, comparing an SVM, three open language models in different sizes, and GPT-3.5. Our results show the challenging nature of the task, with no approach exceeding an F1-score of 0.54. The best-performing model, AGBert, is also available for download3. While the fine-tuned models often performed better with regard to precision, GPT-3.5 outperformed the other approaches with regard to recall. An analysis of the errors indicates that one of the main challenges could be the correct interpretation of complex clauses, rather than the decision boundaries of what is permissible and what is not. The code that was used to prepare the datasets and train the models is also available on GitHub. We hope that the corpus will contribute to enabling future open and reproducible NLP research.

2 Related Work

Legal tasks and datasets have become increasingly relevant for the evaluation of language models over the past years. From domain-specific models, like LEGAL-BERT (Chalkidis et al., 2020), to benchmark datasets, like LexGLUE (Chalkidis et al., 2022), culminating in the presentation of GPT-4 by OpenAI, where the alleged “legal skills” of the model have been one of the main communication points to show its improvement from the previous version (Achiam et al., 2023). Additionally, the question of whether a GPT model would be able to get through law school (Choi et al., 2021) or pass the bar exam (Katz et al., 2023) has been raised repeatedly in the last few years.

2.1 Legal Datasets

In general, there is a large number of legal data sets available. Mostly because many governments and institutions publish laws, court decisions, and similar legal documents digitally and under open licenses. However, only a small fraction of them has been manually annotated (Braun, 2023). Many supervised NLP tasks in the legal domain therefore rely on corpora that inherently contain labels or where labels can be automatically derived. That is, for example, the case for translation where parallel corpora of the European Union can be used (Skadin, š et al., 2014) or outcome prediction for court cases, where the final verdict can be easily extracted from the decision document (Chalkidis et al., 2019). Document types that are not regularly published by governments or institutions, like (consumer) contracts, are rare to find in corpora.

Manual annotation by legal experts is very expensive and therefore rare. While a number of such data sets exists (see Section 2.1.1 and 2.1.2), particularly in English, we are not aware of any corpora in different languages that have a size comparable to the AGB-DE corpus, which consists of 3,764 legally annotated clauses from 93 contracts.

2.1.1 Consumer Contracts

With 100 English Terms of Services and a total of 1,715 annotated clauses, the corpus provided by Ruggeri et al. (2022) is one of the biggest of its kind. Clauses are classified into fair and unfair clauses and among the unfair clauses five major categories are distinguished. Drawzeski et al. (2021) presented a similar sized corpus of 100 Terms of Services, however, consisting of only 25 distinct contracts each of which is available in four lan-guages (English, Italian, German, and Polish).

Another large corpus of consumer contracts is the OP-115 Corpus by Wilson et al. (2016) which consists of 115 privacy policies from websites that have been annotated by law students with data practices

that occur in the text. The MAPP corpus is an even larger privacy policy corpus with 155 privacy policies from mobile applications (Arora et al., 2022), which have been annotated in a similar fash-ion. Both contain only English contracts.

2.1.2 B2B Contracts

In the space of commercial (business-to-business) contracts, larger datasets are available. The CUAD dataset by Hendrycks et al. (2021) consists of 510 English commercial contracts in which 41 different types of legal clauses have been annotated. Chalkidis et al. (2017) provide a data set of 993 contracts that have been labeled with clause headings and 2,461 contracts that have been annotated with other types of contract elements by law students. Unlike, for example, the data set by Ruggeri et al. (2022), but also the AGB-DE corpus, these datasets were not annotated with a legal judgment but rather with regard to the topic of individual clauses.

2.2 Legal Assessment of Contract Clauses

Some of the above-mentioned data sets, but also other, mostly non-published data sets, have been used to train different NLP models in order to predict whether a given contract clause is valid or not. This is also the task we mainly had in mind when we built the AGB-DE corpus. Ruggeri et al. (2022) used a Memory-Augmented Neural Network to detect unfair clauses in Terms of Services and reported an accuracy of 0.526. Braun and Matthes (2021) used a fine-tuned BERT model to predict void clauses in Terms and Conditions of online shops and reported an accuracy of 0.9. Torre et al. (2020) used an SVM to detect missing clauses in privacy policies and reported a precision of 0.85 and a recall of 0.96. Most recently, Martin et al. (2024) presented a study that compares the performance of, among others, GPT4-32k with junior lawyers in locating legal issues in contracts. They report that GPT4 is not only faster and cheaper but also better (F1-score of 0.74) than junior lawyers (F1-score of 0.667).

3 Corpus

The corpus consists of 3,764 clauses from 93 standard form consumer contracts. In this section, the construction of the corpus is described. A detailed datasheet (Gebru et al., 2021) for the corpus can be found in Appendix F.

3.1 Data

The data was collected in 2021 and 2022 and annotated between 2021 and 2023. It consists of German4 standard form consumer contracts that are available online, such as Terms and Conditions of online shops, fitness studios, and telecommunication providers. The data was collected by the annotators, consumer protection lawyers (see Section 3.2 for more details), based on their professional interests. Each clause from each contract was manually copied into an Excel file together with the title of the clause and the URL to the contract text. In total, 93 contracts have been collected in that way.

3.2 Annotation

The dataset was annotated by five fully-qualified lawyers from two German NGOs with a focus on consumer protection. Each of the annotators had multiple years of experience in consumer protection law and consumer counseling. The annotations were made in an Excel file, which annotators preferred over dedicated annotation tools.

3.2.1 Topics

First, one or multiple topic labels were added to each clause. Subsequently, subtopic labels could be added to further specify the content of a clause. For this classification, we used the taxonomy introduced by Braun and Matthes (2022), which consists of 23 topic labels and 37 subtopic labels (see Appendix A

for a list of the available labels). While each clause had to be annotated with at least one topic label, the annotators were instructed to only add subtopics where they found it fitting.

3.2.2 Validity

Afterwards, each clause was legally assessed by the expert annotators. A clause in a contract is void, i.e. cannot be enforced by the parties of the contract, if it contradicts governing law. Whether a clause is actually void depends on many things, including, in some cases, whether one of the parties is a consumer or whether both parties are businesses. The final decision on whether a specific clause in specific circumstances is actually void can only be made by a court of law. Therefore, the instruction for the annotators was to label a clause as potentially void, if they think a consumer residing in Germany could successfully challenge the clause in court. For the remainder of this paper, if we say a clause is void, we mean that it was annotated as potentially void by the expert annotators. In addition to the assessment itself, which is binary (1 potentially void, 0 valid), the annotators can add a comment in the Excel file explaining their decision. By the wish of the annotators, these explanations are not part of the published corpus. It is also worth re-highlighting that the annotators work for NGOs that are dedicated to advocate for consumer rights and their interpretation of the law might therefore be more consumer-friendly than a lawyer who works for a big corporation.

Initially, a small subset of the data was annotated by two experts (one from each organization). In this initial annotation, the legal annotations were in agreement in 76% of the cases. Based on these initial annotations, the experts discussed and aligned their annotation strategies to increase the consistency of the annotations. During this process, two patterns underlying the disagreement became apparent. All disagreements that were found in this phase were based on a disagreement about the interpretation of laws or court rulings, rather than a disagreement about the interpretation of a clause, i.e. the text of the contract. The second pattern was that many of the disagreements were based on laws that use vague legal terms. Laws are often formulated vaguely on purpose. The German Civil Code for example deems clauses void that provide “unreasonably long payment deadlines” (§308 No. 1a). What might seem like bad law-making is done to make laws “future proof”. Whether a payment deadline is unreasonably long, for example, changed significantly between the 1970s, when letters and bank transfers still took multiple days and today. It is up to courts to interpret these terms and these interpretations can also change over time. To increase the alignment of annotators, the annotations guidelines were extended with agreed interpretations of relevant legal provisions that contain such vague legal terms. Additionally, as a “catch-all” solution, a rule was introduced that when in doubt, e.g. because different courts ruled differently, the annotators will always use the more consumer-friendly interpretation. For the subsequent annotation of the complete corpus, each instance was annotated by one annotator. While it would have been preferable to have multiple annotators per instance, that was not feasible due to cost reasons. The total costs of the data collection and annotation for the AGB-DE corpus were approximately 110.000 EUR.

3.3 Anonymization

For ethical and legal reasons, we decided to anonymize the dataset before publishing it. While the original data does not contain any information from individuals, it does contain (publicly accessible) information from companies, like phone numbers, addresses, and tax IDs. We took multiple steps to remove this data from the contracts in order to make it harder to identify the company that drafted the contract. Companies can and do change their contracts over time, so we want to avoid consumers finding and reading an outdated version of a contract. Additionally, it is not unlikely that one or multiple

assessments made by the annotators would not hold up in a court of law, either due to an unconscious mistake or due to the aforementioned bias with regard to the interpretation of the law. Wrongfully claiming a company uses void terms can potentially be harmful to their business and could implicate liabilities. The other way around, wrongfully claiming a clause is valid could potentially harm consumers if they rely on that assessment. Therefore we implemented ten anonymization steps:

1. Remove all clauses with the topic label party from the corpus (these clauses consist only of information about the contracting party, i.e. the company)

2. Replace all email addresses with "hello@example.com" using regular ex-pressions (regex)

3. Replace all URLs with "www.example.com"

using regex

4. Replace all international bank ac-count numbers (IBANs) with "DE75512108001245126199" using regex

5. Replace all tax IDs with DE398517849 using,regex

6. Replace all phone numbers with 00 00 12345678 using regex

7. Replace all ZIP codes with 00000 using regex

8. Replace all names of companies and organi-zations with "«NAME»" using Named Entity Recognition (NER)

9. Replace all city names with "«STADT»" (Ger-man for city) using NER

10. Replace all street names with "«STRASSE»" (German for street) using NER

While the first seven steps turned out to work very well and straight-forward (in total 84 party clauses have been removed and 120 email addresses, 231 URLs, 2 IBANs, 117 phone numbers, and 279 ZIP codes have been replaced), many of the available standard NER libraries turned out to not work very well for the texts. In the end, the FLAIR library (Akbik et al., 2019) with the ner-german-legal model (Leitner et al., 2019) turned out to be most suitable. With the help of the model, we were able to replace 724 names of organizations and companies, 418 city names, and 53 street names. However, a manual inspection revealed that an additional, manual, anonymization round was necessary. In this manual process an additional 1,338 company names, 38 city names, and 85 streets have been removed.

In order to avoid over-anonymization which could potentially result in decreased classification performance, a list of organizations and URLs were explicitly excluded from being removed or replaced. The list mainly included political bodies like the European Union and their URLs, shipping companies and their URLs, and payment provider and their URLs. An excerpt from the final corpus is shown in Table 1.

4 Corpus Analysis

The corpus consists of 93 contracts with 3,764 clauses (an average of 40 clauses per contract), which contain 11,387 sentences (avg. of 3 sentences per clause) and 250,859 tokens (avg. of 22 per sentence). Out of the 3,764 clauses present in the corpus, 179 (or 4.8%) have been annotated as potentially void. That is comparable, although slightly lower, than the 6% reported by Braun and Matthes (2021) on a much smaller dataset of 24 contracts. While that results in a corpus that is imbalanced, we believe it to be a realistic reflection of reality, where void clauses are also significantly less frequent than void clauses.

Id	Con.	Lang	Title	Text	Topics	Subtopics	Void
10	1	de	2. Widerrufs-belehrung	Sie tragen keine Kosten für die Rücksendung der Ware.	withdrawal	withdrawal: shipping-Costs	0
124	3	de	11. Gel-tungsbedin-gungen der AGB	Anstelle der unwirk-samen Vorschrift gilt eine Regelung, die der mit der unwirksamen Vorschrift verfolgten wirtschaftlichen Zweck-setzung am nächsten kommt.	severability		1
127	4	de	1. Allge-meines	1.4. Mit der Bestellung auf dieser Website bestätigen Sie, dass Sie volljährig und rechtsfähig sind, einen Verbrauchervertrag abzuschließen."	age		0
215	9	de	§6 - EIGEN-TUMSVOR-BEHALT	Die von «NAME» gelieferte Ware verbleibt bis zur vollständigen Bezahlung Eigentum von «NAME»	ret.OfTitle		0

Table 1: Excerpt from the corpus

Table 2 shows how the clauses are distributed among the topics and the percentage of potentially void clauses in each topic. Since a clause can belong to multiple topics and subtopics, the sum of the labels is greater than the number of clauses. 30 clauses have been annotated with more than one topic. Out of those 30, one clause has been annotated with three topics and one clause has been annotated with four topic labels, the other 28 have been annotated with two topic labels. Out of the 3,764 clauses, only 1,078 have been annotated with a subtopic. Partially, we believe that to be the result of the fact that it was not mandatory for the annotators to add a subtopic and that the focus of the annotation was clearly on the legal assessment. In total, the corpus contains 8,582 labels.

An analysis of the distribution of void clauses in relation to the topic shows that there are some classes which are particularly prone to be seen as potentially void by the experts. A deeper analysis, together with the experts, revealed that these are very often types of clauses that are particularly strictly regulated. The class changes, for example, captures clauses that relate to changes that are made to the contract after it came into force. Given that standard form contracts are already considered to “reflect an imbalance of contracting power” (Braun et al., 2019), it is not surprising that the possibilities for a company to change them after the customer agreed are very strictly regulated. Therefore, it makes sense that such clauses are exceedingly considered void by the annotators. Similar reasoning can be applied to other topics, like liability or severability clauses.

5 Automated Legal Assessment

In order to present a baseline for the main task for which the corpus was designed and evaluate the difficulty of the task, we compared different language models and an SVM for the classification of clauses

into (potentially) void and valid.

5.1 Data Split

To do so, we created a dataset from the corpus, which is split into a training set of 80% (3,004 clauses) and a test set of 20% (755 clauses)5. We stratified the data split by both the topic labels and the legal assessment to guarantee an equal representation of each class in both datasets. Five clauses from the original corpus were removed in this dataset because they were the only void instances of their clause type in the corpus, and it was, therefore, not possible to split them in the above-described fashion.

Label	Amount	Void (%)
age	20	0.00
applicability	148	2.03
applicableLaw	87	3.45
arbitration	97	1.03
changes	9	11.11
codeOfConduct	29	0.00
conclusionOfContract (cOc)	557	5.92
cOc:binding	50	0.00
cOc:changeOfOrder	1	0.00
cOc:definition	39	0.00
cOc:restrictions	13	0.00
cOc:steps	59	0.00
cOc:withdrawal	17	0.00
contractLanguage	41	0.00
delivery	475	7.16
delivery:brokenPackaging	19	0.00
delivery:costs	53	0.00
delivery:customs	2	0.00
delivery:destination	18	0.00
delivery:methods	26	0.00
delivery:partial	22	0.00
delivery:time	1	0.00
description	46	0.00
disposal	36	0.00
intellectualProperty	39	0.00
language	9	11.11
liability	211	9.00
party	0	0.00
payment	642	6.07
payment:fee	6	0.00
payment:late	27	0.00
payment:loyalty	1	0.00
payment:methods	289	0.00
payment:restraint	5	0.00
payment:vouchers	124	0.00
personalData	115	0.87
personalData:cookies	6	0.00
personalData:duration	0	0.00
personalData:information	1	0.00
personalData:reason	1	0.00
personalData:update	0	0.00
personalData:usage	1	0.00
placeOfJurisdiction	53	1.89
prices	147	1.36
prices:currency	6	0.00
prices:vat	36	0.00
retentionOfTitle	125	2.40
severability	35	11.43
textStorage	57	1.75
warranty	314	6.37
warranty:options	4	0.00
warranty:period	10	0.00
withdrawal	506	3.75
withdrawal:compensation	7	0.00
withdrawal:effects	45	0.00
withdrawal:exclusion	51	0.00
withdrawal:form	25	0.00
withdrawal:model	4	0.00
withdrawal:period	33	0.00
withdrawal:shippingCosts	11	0.00
withdrawal:shippingMethod	7	0.00
Total lvl 1	3798	4.80
Total lvl 2	1020	

Table 2: Distribution of topics and void clauses

5.2 Undersampling

Because it was clear that the dataset would be challenging because of its imbalanced nature, we created a second dataset6, in which we used undersampling (Liu et al., 2008) to remove data from classes that are over-represented. In particular, the number of instances from each combination of topic and validity was limited to 100. I.e., if there were 40 void clauses of one topic and 120 valid clauses of the same topic, only 100 of the valid clauses would go into this second dataset. In this way, we ended up with a dataset that consists of 1,362 (80%) clauses for training and 345 (20%) for testing. The split of the data remained the same, i.e. we only removed data but did not change the distribution of existing data between the training and test sets. For easier distinction, we will refer to the first larger dataset as agb-de and to the undersampled dataset as agb-de-under.

5.3 Models

We used both datasets to train an SVM that uses tf-idf vectors of the clauses as input and fine-tune and evaluate four different language models:

- The BERT model bert-base-german-cased (Chan et al., 2020), which was also trained on the Open Legal Data dataset that consist of more than 100,000 German legal documents (Ostendorff et al., 2020)
- The multilingual RoBERTa model xlm-roberta-base (Conneau et al., 2019)
- The German GPT2 model gerpt2 (Minix-hofer, 2021)
- And finally gpt-3.5-turbo-0125, which was evaluated in a “zero-shot” fashion without fine-tuning

For the fine-tuning of the three open models, we conducted a manual hyperparameter search, starting with the standard hyperparameters for each model. The final hyperparameters that were used for the fine-tuning can be found in Appendix B, as well as in the training code that is published alongside the corpus. In order to further address the imbalance of the dataset, we also used a tailored loss function. For the fine-tuning of all models, we used class weights of 1.0 for valid and 100.0 for void clauses in the loss function. The prompt and API call used for the evaluation of GPT-3.5 can be found in Appendix C.

6 Evaluation

The results of the evaluation are shown in Table 3. The main metric that we focused on for the evaluation is the F1-score, which provides a balance between the two unequally distributed classes.

For the agb-de dataset, none of the models was able to handle the very imbalanced dataset well. The best-performing model on the dataset was the fine-tuned BERT model. While RoBERTa is usually better than BERT in handling imbalanced data (Younes and Mathiak, 2022), the RoBERTa model that we fine-tuned is a multilingual model, while the BERT model that we fine-tuned was specifically pre-trained on German legal data. The second best performance was achieved by the SVM with an F1-score of 0.31. For GPT-3.5, there was not much difference in the performance independent of the dataset: For both datasets, GPT-3.5 achieved the highest recall, at the cost of a very low precision. I.e., GPT-3.5 falsely classified the majority of clauses as potentially void (see also Figure 2d).

All models performed better on the agb-de-under dataset, showing that undersampling is a suitable strategy for the corpus. BERT again performed best with an F1-score of 0.54. To test whether the improvement performance on the undersampled dataset was caused by a better model and not just by the fact that, by chance, difficult items got removed from the test set, we also tested all models trained on the undersampled dataset on the original test set from the agb-de dataset. The results of this evaluation are shown in Appendix D. Except for the SVM, all models performed better when being trained on the undersampled dataset and evaluated on the full dataset compared to being trained and tested on the full

dataset, indicating that removing excessive imbalanced training data can indeed lead to better models on this dataset.

7 Error Analysis

Figure 1 shows precision and recall for each model on the undersampled dataset for the largest topics in the corpus. The numbers show that the models perform very differently for the individual topics. For liability, for example, no model was able to achieve a recall higher than 0.25. With 11% of void clauses, liability clauses are most frequently void among the large topics. For the expert annotators, liability clauses were relatively easy to annotate, because of very explicit regulations. § 309 No. 7 of the German Civil Code, for example, explicitly states that "an exclusion or limitation of liability for damage from injury to life, limb or health due to negligent breach of duty by the user" is void. Linguistically, however, an analysis of the liability clauses shows that they are often complicated, with multiple explicit inclusions and exclusions in one sentence, which could be a reason for the poor performance of most models.

While clauses of a certain topic can be void for different reasons, for some topics there are predominant patterns. Warranty clauses, in the corpus, for example, are predominantly void because they restrict the warranty in cases where defects are not reported within a specified (too short) time frame. Similarly, payment clauses are predominantly void because they introduce excessive fees for late payments. The BERT model was better at picking up these more repetitive patterns compared to the more nuanced other topics. The GPT-2 model achieved the highest precision for payment clauses, which have the largest share in the corpus, and therefore heavily influence the overall result. Overall, the performance differences between classes could indicate that using the topic labels as an additional feature could improve classification performance.

The performance of GPT-3.5 differs from the other models as it was more prone to generating false positives, i.e. flagging clauses as void that are actually valid. The prompt we used for GPT-3.5 not only asked it to perform the classification into potentially valid and void but also asked the model to provide an "explanation"7. While the texts provided are not an explanation in the sense that they make the reasons for the assessment transparent, it is still interesting to see that the issues described are most of the time correct, however, the assessment is still wrong, in both the text and the annotation. For a clause about withdrawals (corpus ID 5), GPT-3.5 for example "explains": "The clause is potentially invalid as it could unfairly disadvantage the consumer. According to the law, the consumer must be able to exercise his right ofwithdrawal clearly and conspicuously, without any additional hurdles or conditions being imposed. By specifying here that registering a return under My Account is considered a revocation, this could limit the consumer's ability to exercise their right of revocation. It is important that the consumer can exercise his right ofwithdrawal without additional obligations." (see Appendix E.1 for the original German text) While the text by itself would be factually correct, it is not applicable to the clause in question. The clause clearly describes other ways to exercise the right to revoke a contract and explicitly states that it is not mandatory to use the "My Account" feature. In line with the text, the clause was falsely labeled as potentially void by GPT-3.5.

Dataset	Model	Precision	Recall	F1-score
agb-de	svm	0.37	0.27	0.31
	bert-base-german-cased	0.50	0.27	**0.35**
	xlm-roberta-base	0.00	0.00	0.00
	gerpt2	**0.71**	0.14	0.23
	gpt-3.5-turbo-0125	0.06	**0.92**	0.11
agb-de-under	svm	0.40	0.32	0.36
	bert-base-german-cased	0.51	0.57	**0.54**
	xlm-roberta-base	**0.75**	0.08	0.15
	gerpt2	0.64	0.43	0.52
	gpt-3.5-turbo-0125	0.13	**0.92**	0.22

Table 3: Results of the evaluation, best performance on a dataset is highlighted in bold

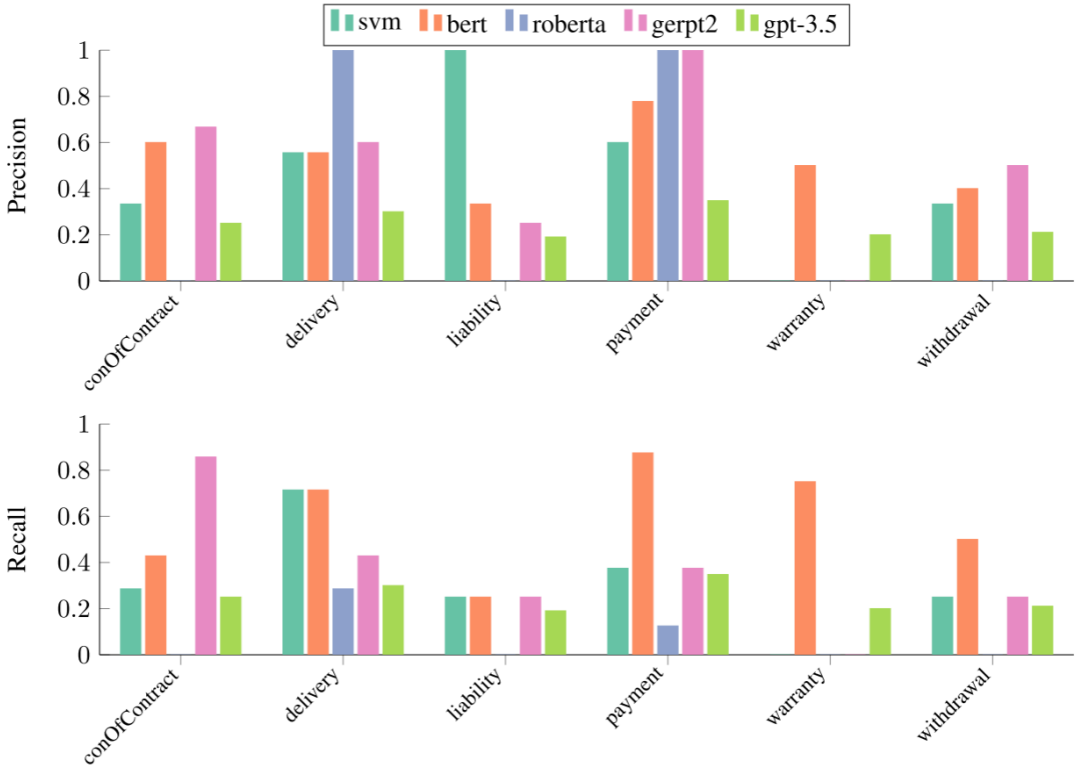

Figure 1: Precision and recall per topic on the agb-de-under dataset

Actual \ Predicted	valid	void
valid	288	20
void	16	21

(a) bert-base-german-cased

Actual \ Predicted	valid	void
valid	307	1
void	34	3

(b) xlm-roberta-base

Actual \ Predicted	valid	void
valid	299	9
void	21	16

(c) gerpt2

Actual \ Predicted	valid	void
valid	71	237
void	3	34

(d) gpt-3.5-turbo-0125

Figure 2: Confusion matrices for the evaluation on the agb-de-under dataset

Similarly, for the clause “If the customer is a merchant [...] the place of jurisdiction for all disputes arising from this contract is the court responsible for the seller’s place of business [...].” (corpus ID 3212)

GPT-3.5 concludes the clause is potentially void, because "The clause puts the consumer at an unreasonable disadvantage because this clause binds the consumer to the company's place of jurisdiction''. The legal reasoning of that is again correct, however, not applicable to the clause, because the clause only applies to merchants in which case it is valid. To test whether GPT-3.5 is aware that the clause is valid for merchants, we adapted the prompt to say that the system should imagine being a lawyer for a merchant (instead of a consumer) and indeed the response was "The clause is unlikely to be potentially invalid if the customer is an entrepreneur." More often than not, the main challenge for GPT-3.5 seems not to be missing legal "knowledge", but the incorrect interpretation of the clause, especially in cases where the clause contains elements that are optional or not applicable to all customers.

Another typical problem of LLMs in general can be seen in the explanation generated for the clause with the corpus ID 10. Here, the model generates the following text: "According to Section 357 of the German Civil Code (BGB), a consumer may not be charged the costs of return shipping in the event of a revocation. Therefore, a clause requiring the consumer to bear the costs of return shipping is potentially invalid." (see Appendix E.2). Until 2014, it was indeed the case that "a consumer may not be charged the costs of return shipping in the event ofa revocation" (at least for purchases over 40 EUR). However, today, §357 BGB explicitly states the opposite, i.e. that customers have to bear the costs of return shipping. However, texts that relate to the old legislation are probably more frequent in the data GPT-3.5 was trained on, which is data up to 2021.

8 Conclusion

In this paper, we have introduced the AGB-DE corpus, consisting of 3,764 clauses from German consumer contracts that have been annotated by legal experts. In addition to the corpus, we presented two datasets that have been derived from the corpus and have been split into training and test sets. The datasets can easily be used to train or fine-tune machine learning models. We have used these datasets to provide a benchmark on the corpus for the task of classifying whether a clause is valid or potentially void.

For the dataset that is representative of the distribution in the corpus, we showed that language models struggle with the imbalanced data and could not achieve an F1-score above 0.35. For the second dataset, which is more balanced through under sampling, we showed that open models like BERT and GPT2 were able to better identify void clauses achieving an F-1 score of 0.54 and 0.52. On both datasets, open models outperformed GPT-3.5 with regard to F1-score, which generated a huge amount of false positives leading to an F1-score of 0.11 on the more imbalanced and 0.22 on the less imbalanced dataset.

Ethics

False legal statements always have to potential to cause harm. While we worked with experienced experts who carefully decided on each annotation, it is always possible that the dataset contains errors. Claiming falsely that a clause is void could potentially have negative impacts on a business, claiming falsely that a clause is valid could potentially have a negative impact on consumers. One measurement we took to avoid such impacts is anonymising the clauses, in order to make it harder to connect them with a specific business. Many clauses, like severability or liability clauses, are highly standardised and often directly drawn from boilerplate contracts. In such cases removing the explicit identifiers is sufficient to make them completely anonymous. In other cases, particularly for example in clauses about bonus and rewards programs, even after removing the explicit identifiers, clauses can be so specific that they can be traced back to individual businesses, if they still use the same clause. We believe that the practical impact of that is rather limited, because retrieving the information about a specific clause from the dataset is not

something that a significant number of potential customers is likely going to do. We added an explanation text to the readme of the published version of the corpus to highlight that the annotations are based on the perspective of individual experts and not based on a concrete court ruling and therefore not legally binding. Overall we believe that the publication of the dataset can help to address an existing imbalance of power between customers and companies and thereby, together with the anonymisation strategy, warrants the small but existing risk.

At the same time, being aware that a company uses potentially void clauses that disadvantage consumers and not doing anything about it could be seen as unethical. Therefore, the NGOs we worked with did take legal steps if they encountered clauses during the annotation that they deemed so disadvantageous for consumers that legal steps were necessary.

In general, the dataset consists of publicly accessible data that is usually carefully drafted by lawyers. We did not encounter any instances within the dataset that we deem problematic from an ethical perspective.

Limitations

Due to practical restrictions, the presented dataset has several limitations:

• The dataset was annotated from a consumer protection perspective and therefore is most likely biased towards interpreting existing regulations in a consumer-friendly way.

• The instances in the dataset were only annotated by one expert. Legal decision-making involves uncertainty and interpretation, therefore it would have been desirable to have each instance annotated by multiple experts.

• The assessments in the corpus are based on the legal regulations at the time of the annotation(2021-2023), models trained on newer data might correctly make different predictions based on legislative changes or new court decisions.

• While we believe the dataset to be a somewhat representative mapping of the real world, that also means that is imbalanced and contains a relatively small amount of void clauses. Due to practical restrictions, the presented evaluation has several limitations:

• Using cloud-based models like GPT-3.5 always poses a threat to the reproducibility of the results, through using an explicitly versioned instance of the model we try to minimize the risk.

• Arguably comparing models that were finetuned on a specific task with a zero-shot prompt approach is an unequal comparison. Future strategies to improve the GPT-3.5 performance could include using few-shot approaches or also prompt engineering. While, given the error results, the few-shot approach might have limited effect, we believe that prompt engineering might be effective in reducing the large number of false positives.