律师事务所行政管理智能助手系统的设计与实现

摘 要：随着法治的不断进步，人民对法律服务的质量、效率不断提出新的要求，律师事务所在为人民提供法律服务的过程中也遇到了前所未有的挑战，为满足客户对简单法律问题的咨询需求，缓解行政前台的接待压力，保证律所的管理秩序，本文设计了一款律师事务所行政管理智能助手系统。本系统的应用可使律所管理有序化，有助于提高律所行政前台的办公效率，它可作为律所的网络宣传窗口，有利于提升律所的品牌建设可实现客户的有效管理，方便律所与客户的交互，及时获取客户的反馈。

通过分析系统需求和可行性等因素，遵循标准的软件开发流程，本文完成了该系统的设计、开发和测试。本文利用网络爬虫在华律网等法律网站爬取问答数据集，再进行数据的预处理，采用Seq2Seq结合Attention机制训练模型实现了智能咨询模块，它可以方便客户在线咨询常见的法律问题；通过Flask框架实现了值班律师、注册登陆、预约办事、关于律所、联系反馈、后台管理等功能模块，从而实现了律所与客户之间对接窗口的搭建；使用SQLite存储咨询数据集，MySQL数据库存储用户和律所数据，从而实现数据的管理。整个系统功能完善、界面友好，具有较好的应用价值。

关键词：法律助手；Seq2Seq；法律咨询；律所管理

Design and implementation of intelligent assistant system for law firm administration

Abstract**:**With the continuous progress of the rule of law, people constantly put forward new requirements for the quality and efficiency of legal services, and law firms have also encountered unprecedented challenges in the process of providing legal services for the people. In order to meet the clients' demand for consulting simple legal issues, ease the reception pressure of the administrative reception, and ensure the management order of the law firm, this paper designs an intelligent assistant system for the administrative management of law firms. The application of this system can make the management of the law firm orderly, help to improve the office efficiency of the administrative reception of the law firm. It can be used as the network propaganda window of the law firm, help to improve the brand construction of the law firm, realize the effective management of customers, facilitate the interaction between the law firm and customers, and obtain the feedback of customers in time.

By analyzing the system requirements and feasibility and following the standard software development process, this paper completes the design, development and testing of the system. This paper uses web crawler to crawl the question and answer data set in the legal website such as China law website, and then preprocess the data. We use Seq2Seq combined with Attention Mechanism training model to realize the intelligent consultation module, which can facilitate customers to consult common legal issues online; Through the Flask framework, the function modules of on duty lawyer, registration and login, appointment, about the law firm, contact and feedback, background management are realized, so as to build the docking window between the law firm and customers; SQLite is used to store consulting data sets, and MySQL database is used to store data of users and law firms, so as to realize data management. The whole system has perfect function, friendly interface and good application value.

Key words**:** legal assistant; Seq2Seq; legal consulting; law firm management

1 绪论

1.1 选题的背景及意义

时至今日，人们对法律服务的需求日益增长，而最重要的法律服务提供平台——律师事务所，其传统的服务模式在这个互联网信息技术高度发达的时代却略显疲态。观察目前我国律师事务所的数量仅为3万余所，且仍然存在发展规模较小、缺乏系统管理等问题。律所业务分布表现为证券、金融、兼并、知识产权等高端的专业性业务集中分布于北京、上海等城市；而传统的民事业务在经济不发达的地区竞争尤为激烈。在此情况下，律所应当谋求多元化，专业化，均衡化，高效化发展，运用科技的力量，挣脱传统服务方式的枷锁，以满足客户需求[1]。

律师事务所行政前台，身为律所与客户沟通交流的第一窗口，肩负着许多任务，主要包括：预约办事，安排值班律师，回复简单法律问题等。结合互联网技术的新型服务方式对律所具有重大的意义：法律服务信息化、智能化是律所发展的时代要求。处理事务规范化、高效化是律所业务拓宽的必备条件。网络推广是律所打开知名度的绝佳方案。高效便捷地解决法律问题是用户的必然需求[2]。人工智能发展至今应用于法律领域不仅具有必要性，还具有可行性，无论是智能咨询，还是解决纠纷、法律检索、文书处理，都体现出了相当的优势[3]。由此，设计一个面向律所和客户之间沟通的窗口，既可以作为宣传律所的网络渠道，缓解行政前台的工作压力，又可以便利民众咨询常见的法律问题，无疑会成为法律服务智能化、系统化的新角度。

1.2 国内外研究状况

2016年，“智慧法庭”被纳入中国国家发展战略，2017年国务院颁布的《新一代人工智能发展规划》中又明确提出了人工智能+法律[4]。法律服务逐渐趋于智能化，人工智能辅助立法系统、智能司法系统等技术产品陆续在人们的视野中出现，极大地推动了立法、司法的智能化进程，为人们的日常生活、法律工作提供了极大的便利[5]。但是目前对于律师事务所的对外行政管理、智能化咨询等方面的探索相对薄弱。

律师事务所的管理信息化系统大多是通过辅助律师办公，将案卷信息化，管理律师信息等面向律所内部管理的产品[6]。如：基于简单局域网共享法律文本和资料，单机版的桌面应用程序，基于CS或BS架构的律所内部管理系统[7]。

目前的问答系统主要集中在开放领域，解决的问题大多是衣食住行等生活问题，如私人购物助理：阿里小蜜等，在法律领域的问答系统研究仍有较大的空白[8]。而法律咨询系统也多为论坛形式[9]，即通过留言的方式提问，需要长时间的等待[10]。

基于深度学习的问答模型在回答效果上相比于传统的问答模型具有更显著的优势[11]。研究表明，序列到序列（Sequence to Sequence，Seq2Seq）模型的生成式自动问答系统推理能力突出，有很大的发展潜力[12]。目前Seq2Seq模型也是最常用于聊天机器人的深度学习模型[13]。

多国学者已经涉入人工智能与法律的研究，不仅以案例为基础的英美法国家重视人工智能在立法和司法中的应用，传统的大陆法系国家，如：德国、法国、日本也进行了相关的研究探讨[14]。法律服务系统在国际社会也同样得到了广泛的应用，如：美国的Ross Intelligence用于解决破产法率问题，英国的在线法院用于化解纠纷，DoNotPay平台用于普及日常法律知识等。

国外律师事务所的信息化经历了开始、增长、发展和成熟四个阶段，已经实现了从金字塔向扁平型的转变，如：美国80%律所实现了信息化办公。并且国外的律师事务所还在不断更新网络化法律服务的方式。例如：英国安理律师事务所的Deal Room，在保证客户信息安全的情况下，便利了客户与律师之间的交流[15]。

1.3 主要工作及研究内容

本文主要基于现实法律服务需求，设计并实现了律师事务所行政管理智能助手系统，用于智能回复用户简单法律问题，推广律所，办理律所对接客户的行政事务。本文智能咨询模块的目标是通过基于深度学习的自然语言处理（Natural Language Processing，NLP）技术实现对常见法律问题的智能问答；值班律师、注册登录、预约办事、关于律所、联系反馈等模块的目标是完成律所与客户之间的信息交流；后台管理模块的目标是实现律所的信息化管理。

1.4 本文的组织结构

根据现实法律服务需求，本文设计并开发了律师事务所行政管理智能助手系统，本文共分为六个章节：

第一章绪论：阐述本文选题的背景和意义，国内外相关领域中的研究状况，有待进一步研究的问题，主要工作，研究内容等；

第二章相关技术：阐述实现本设计将采用的主要技术，包括Seq2Seq模型、注意力（Attention）机制、Flask框架、MySQL、SQLite等；

第三章需求分析：主要阐述本设计的功能需求、性能需求及其他需求等；

第四章系统设计：主要阐述本设计的总体架构、功能模块设计、数据库设计等；

第五章系统实现：主要阐述本设计的实现情况，包括值班律师、注册登录、事务咨询、预约办事、关于律所、联系反馈、后台管理等模块的实现；

第六章系统测试：主要阐述本设计的功能测试、页面测试、性能测试。

2 相关技术

2.1 序列到序列模型

Seq2Seq模型是目前的主流NLP模型，具有广泛的应用，如：问答机器人、翻译、自动生成文本摘要、图片描述等[16]，其结构如图2.1所示。

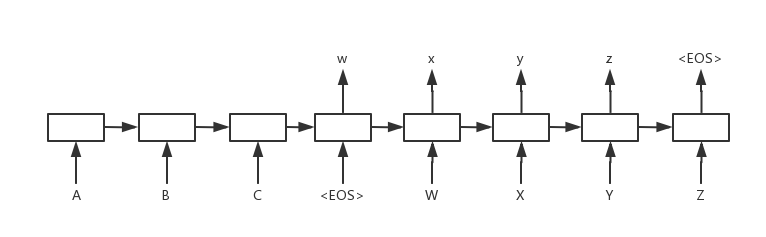


图2.1 Seq2Seq结构图

Seq2Seq由编码器和解码器组成，由于RNN能够自适应输入输出的特性，编码器和解码器一般是由循环神经网络构成[17]。

Seq2Seq的特点在于：输入和输出长度不定；输入输出的各个元素具有关联关系。

单纯的Seq2Seq模型存在弊端，如：句子过长，会出现梯度消失的问题，这是由于使用最后的一个隐藏层输出的向量，越靠近结尾的单词，“记忆”越深刻，越远离结尾的单词，“记忆”越浅薄。面对此等问题，我们需要进一步完善改进。

2.1.1 循环神经网络

RNN在NLP中得到了广泛应用，不同于传统的神经网络模型，每层之间的节点没有连接。RNN隐藏层之间的节点有连接，且隐藏层的下一个输入不仅包括输入层的输出，还有上一时刻隐藏层的输出[18]。如下图2.2所示。

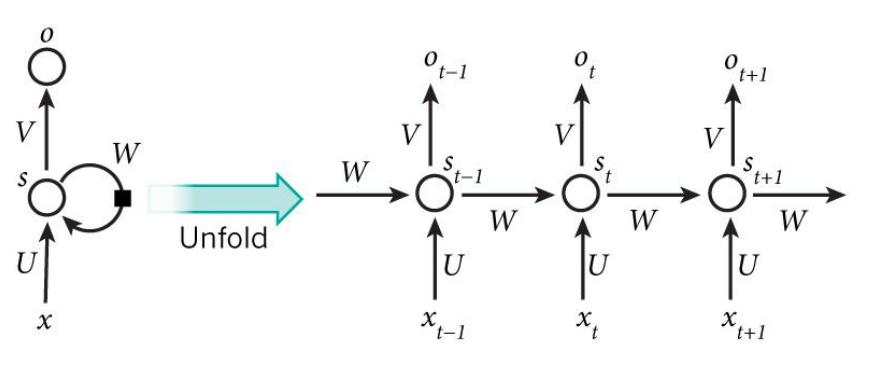


图2.2 RNN结构图

本文采用序列输入和序列输出不同步的RNN，结构如图2.3。

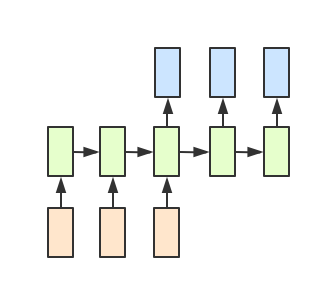


图2.3 many to many结构图

2.1.2 长短时记忆模型

RNN中的长短时记忆模型[(Long Short-Term Memory，LSTM)](https://en.wikipedia.org/wiki/Long_short_term_memory)是目前被广泛使用的模型，该模型比vanilla RNN能够更好地表达长短时依赖。相对于一般的RNN，LSTM通过三个门来实现长时间记忆[19]。即通过三种门更新每个单元模块的状态信息，遗忘门（Forget gate）控制前一个单元传递到当前单元的遗忘程度，输入门（Input gate）和一个tanh函数控制新信息的加入，输出门（Output gate）控制当前单元的信息输出[20]。结构如图2.4所示。

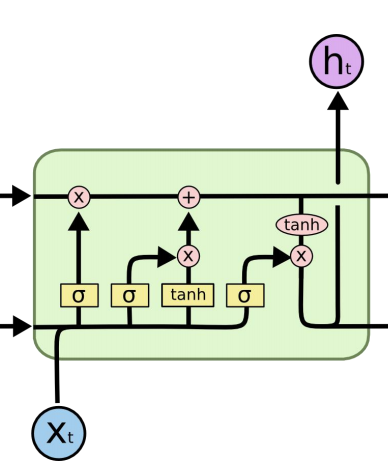


图2.4 LSTM结构图

2.1.3 注意力机制

梯度消失的问题需要一个更复杂的记忆模型，保证只记住真正重要的状态，为此我们引入Attention机制[21]。Seq2Seq模型结合Attention机制的示例如下图2.5。

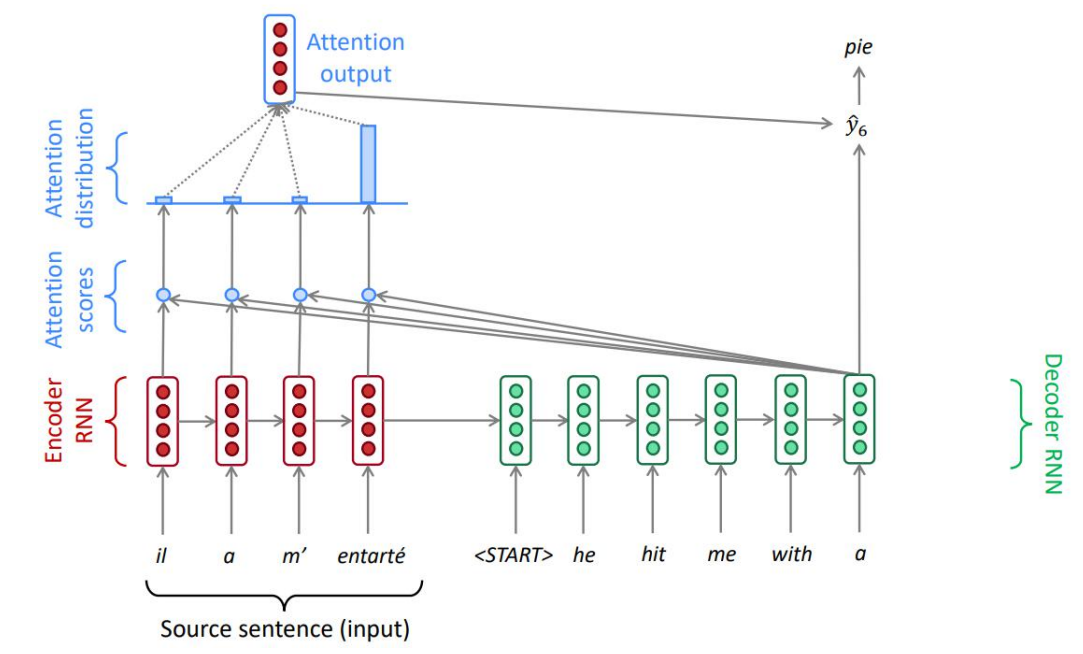


图2.5 Seq2Seq模型结合Attention机制计算示例图

Attention机制多应用于序列学习，其本质就是关注重点，提纲挈领。即模拟人在观察事物时，注重其中重要部分的特性，避免结尾得到重视而前面失去焦点的现象[22]。在Enocoder模块中加入Attention机制，会对输入数据分配权重。

Attention的思路为“带权求和”。首先计算query和key相似度，获得各部分权值；再将权值归一化，得到权重；最后将权重和value加权求和，计算示例如下图2.6。

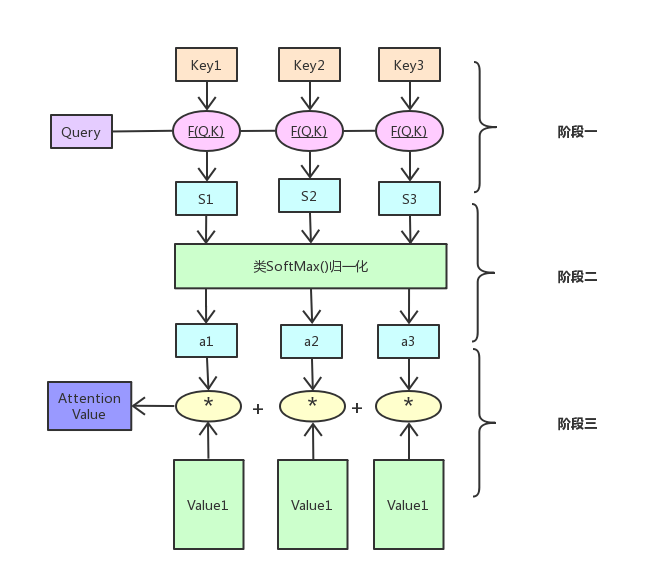


图2.6 Seq2Seq模型结合Attention机制计算示例图

2.2 Flask

Flask——Python编写的轻量级Web应用程序框架，其核心在于“微”，但是“微”并不代表功能弱，而是指保持核心简单而又可扩展。Flask是一个优秀的胶合层，小巧、灵活、定制化程度高，适用于小型网站。Flask本身不包含数据库抽象层、表单验证等，但经过扩展，可以为应用添加这些功能，支持数据库整合、上传处理、表单验证等。扩展包主要有Flask-migrate——管理迁移数据库；Flask-SQLalchemy——操作数据库；Flask-script——插入脚本等。

Flask只依赖两个库Werkzeug和Jinja2，同时Flask也是Werkezug和Jinja2之间的桥梁，Werkzeug出色地实现了PythonWeb服务器网关接口协议（PythonWeb Server Gateway Interface，WSGI），Jinja2用于渲染页面的模板语言。

Werkzeug路由性可以根据复杂度自动为路由排序，即可以以任意顺序声明路由，路由系统仍然能够正常工作；Jinja与MarkupSafe共用，避免渲染页面时不可信的输入，防止攻击注入；ItsDangerous保证数据完整性，用于保护Flask的session cookie；Click用于提供flask命令，并允许添加自定义管理命令。

2.3 MySQL

MySQL——关系型数据库管理系统，将数据保存在不同的表中，提高了操作的灵活性，使用SQL常用访问数据库的标准化语言，具有体积小、速度快等优势。

2.4 SQLite

SQLite——轻量级数据库，由SQL编译器、内核、后端组成，遵循原子性、一致性、隔离性、持久性（Atomicity，Consistency，Isolation，Durability，ACID），使用只需要一个动态库，无需安装和管理配置，约13万行C代码，仅有4.43M，整个数据库（定义、表、索引和数据）都存储在单一的文件中，方便管理和维护，可以跨语言、跨平台向后兼容，还具有速度快，文件格式稳定，安全可靠等优点。

3 需求分析

3.1 系统功能需求

法律智能助手面向寻求基础法律帮助的用户以及律所的行政前台管理人员，用户用例图如图3.1所示，管理员主要管理数据的增、删、改、查，管理员用例图如图3.2所示。

3.1.1 值班律师功能

用户可以查看律所公示的值班律师，以便知悉今日值班律师。

用户可以查看值班律师的详情，以便深入了解值班律师，考虑是否需要值班律师提供法律服务。

3.1.2 注册登录功能

用户通过注册，提交信息，从而获取用户身份。

用户通过登录，从而在网站获取更多信息，享受更多服务。

用户在离开时可以根据意愿，选择退出登录。

3.1.3 事务咨询功能

用户使用智能咨询简单法律问题，进而确认是否需要进一步咨询，委托律师。

用户可以根据常见问题的分类选择常见问题，实现便捷输入，也可以输入自己问题，实现个性化输入。

用户使用人工咨询，以达到快速便捷提问、及时收到反馈的目的。

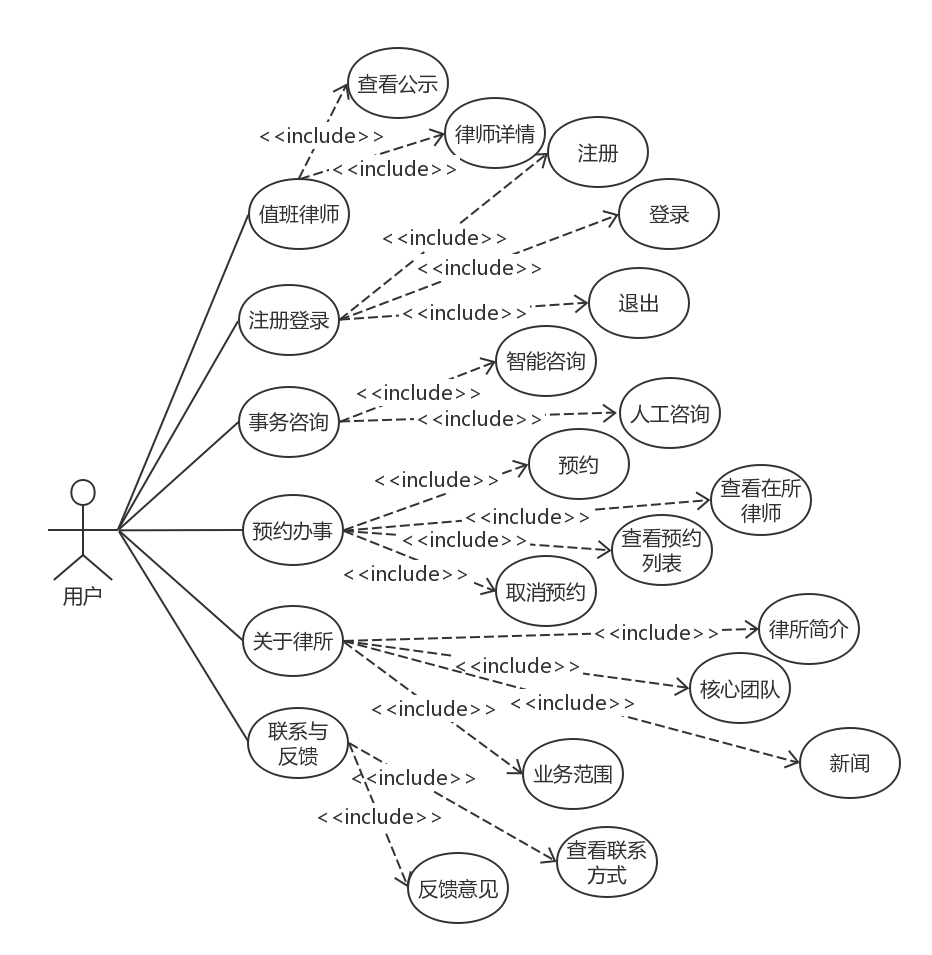


图3.1 用户用例图

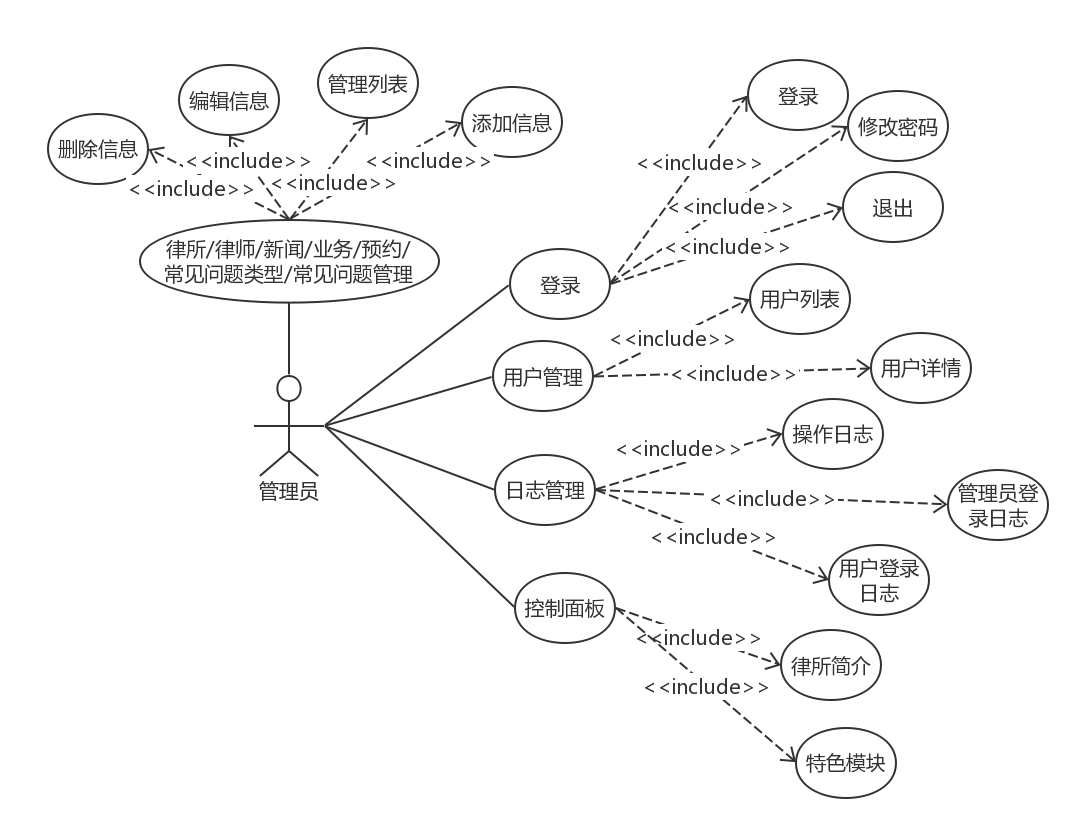


图3.2 管理员用例图

3.1.4 预约办事功能

用户通过预约办事可以查看在所律师并在线预约节约时间，提高办事效率，同时也便于律所系统地管理预约信息，提前做好接待安排，减轻接待压力，亦可避免不必要的骚扰。

3.1.5 关于律所功能

用户通过查看律所的相关信息，包括律所简介、业务说明、核心团队成员、律所的最新动态，以此了解律所强项，进而选择法律服务， 也有利于推广和宣传律所。

3.1.6 联系反馈功能

用户查看各个分所的联系方式，以便寻求法律帮助，也有利于律所的推广和宣传。

用户通过提交意见和建议，与律所进行信息交互，也便于律所改进。

3.1.7 后台管理功能

管理员可以登录、修改密码等，以便登录管理后台。

管理员对数据进行的增、删、改、查等操作，从而管理前台的动态数据，包括管理员管理、用户管理、律所分所管理、新闻管理、业务范围管理、预约管理、日志管理、常见问题类型管理、常见问题管理等。

3.2 系统数据需求

系统主要E-R图包括用户、律师、分所、意见建议等实体，如图3.3。常见问题E-R图包括问题、问题类型等实体，如图3.4。新闻具有id，新闻标题，新闻内容，发布时间等属性。业务具有id，业务名称，业务内容，新增时间等属性。

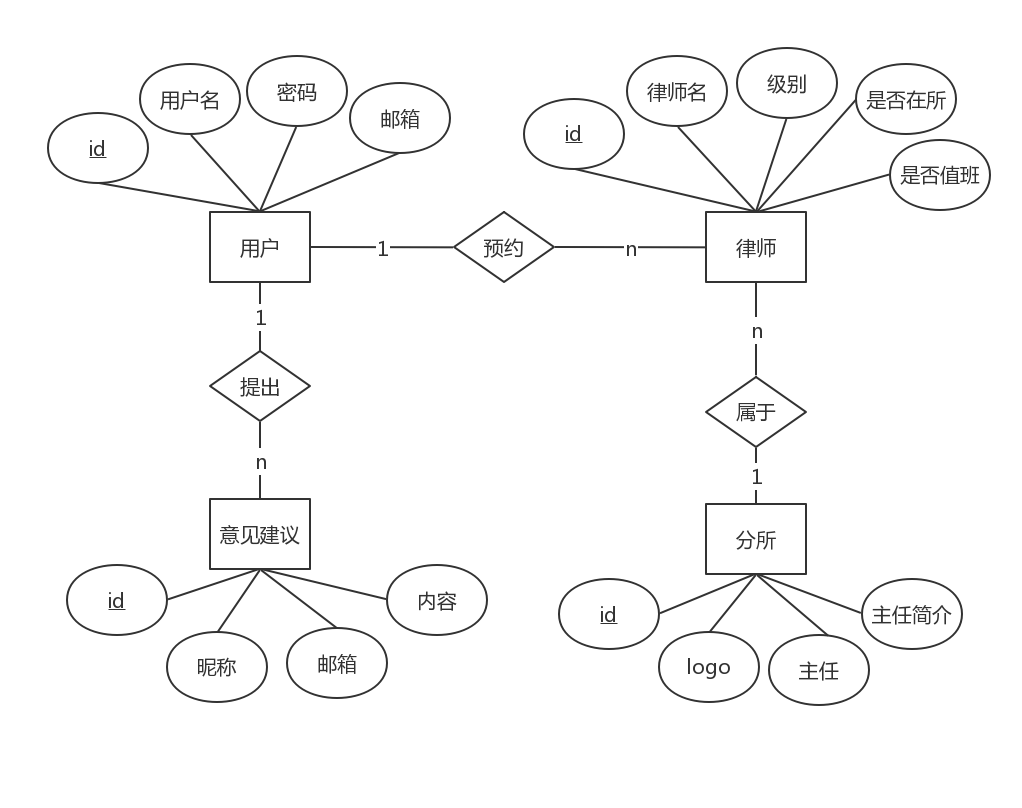


图3.3 系统主要E-R图

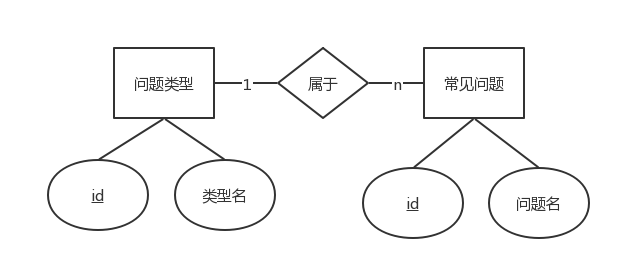


图3.4 问题E-R图

3.3 系统其他需求

本系统能够提供基于Seq2Seq的自动问答功能；系统化管理行政前台对客户的互动交流；系统的整体框架设计良好，采用多层软件架构模式；系统分离功能模块，保证低耦合；提取基类封装，保证系统可重用；系统权限管理完善；界面友善，可操作性强；系统能够健壮、稳定运行；规范化编码，使用SQL标准语言，方便扩充，保证可扩展性；上传文件、用户和管理员密码加密，保证系统的安全性。

4 系统设计

4.1 总体架构

4.1.1 总体功能结构

经过需求分析，系统的前台总体功能总体设计如图4.1所示，后台总体功能设计如图4.2所示。



图4.1 系统前台总体功能结构图

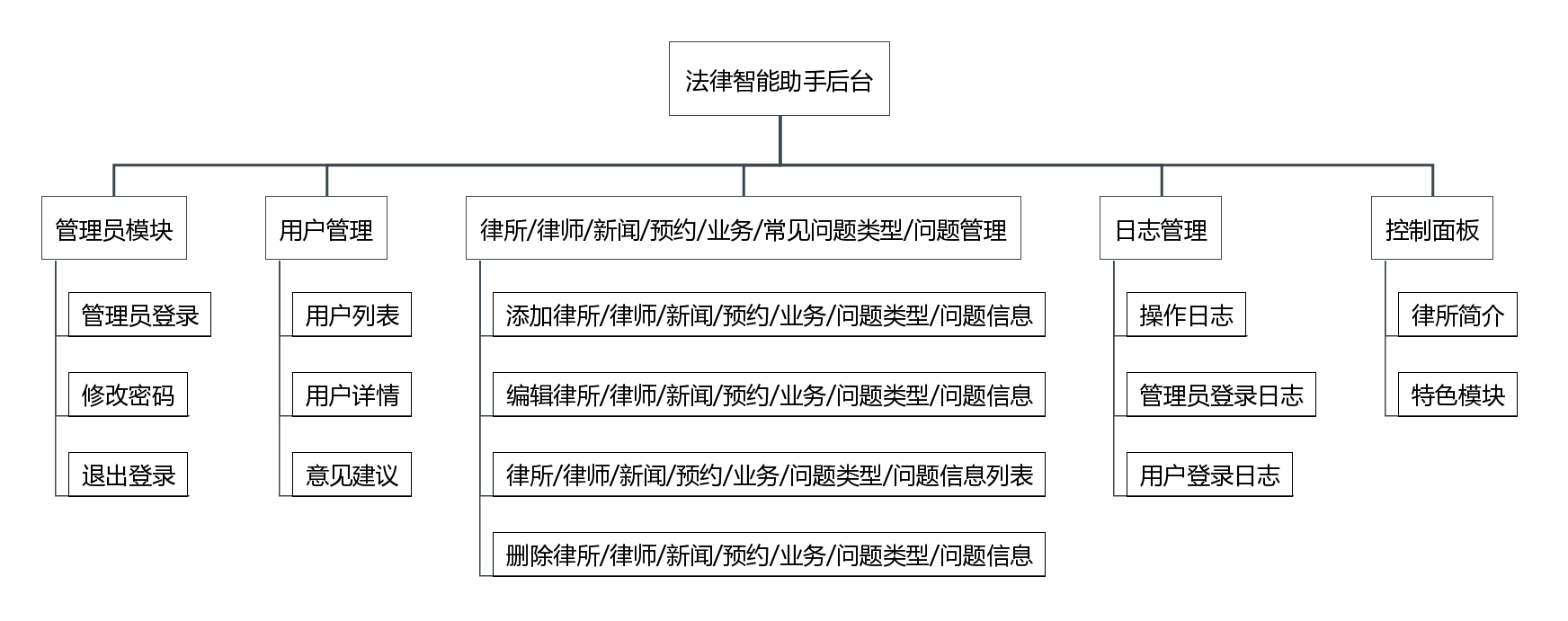


图4.2 系统后台总体功能结构图

4.1.2 系统整体框架

系统整体框架如图4.3所示。

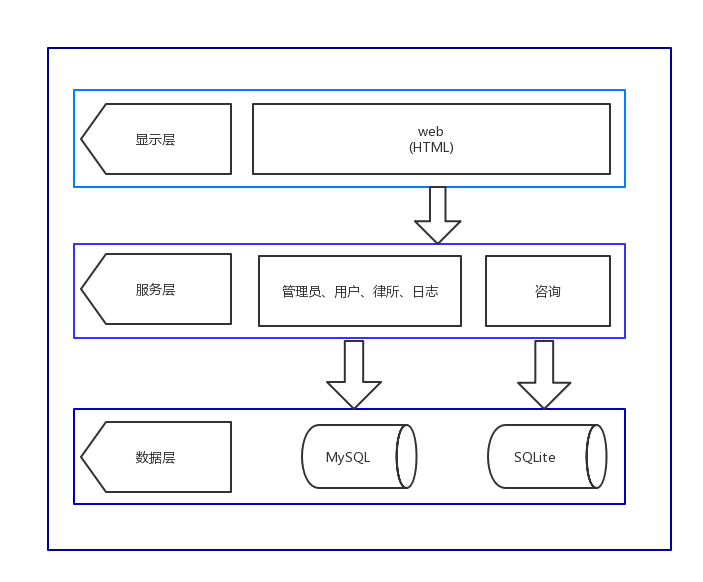


图4.3 系统整体框架图

4.2功能模块设计

4.2.1 前台模块设计

值班律师模块：用于实现用户查看今日值班律师，了解值班律师的详细信息；

注册模块：用于实现新用户注册的功能，在注册页面，用户需填写信息满足以下条件：1）用户名：不能为空；2）邮箱：不能为空，满足邮箱格式，并且每位用户只能使用唯一的邮箱注册；3）密码：不能为空；4）确认密码：不能为空，并且与“密码”保持一致；5）验证码：验证码不能为空，并且正确；

登录模块：用于实现用户登录功能，由于用户的邮箱是唯一的，所以使用邮箱和密码作为登录凭证，在登陆页面填写用户邮箱和密码，单击“登录”即可实现登录；

事务咨询模块：用于实现智能咨询功能，对用户提出的每个基础法律问题，能够自动生成一个回答；常见问题提示，能够将常见问题分类显示，提示用户输入；人工咨询，能够连续回答用户提出的法律问题；

预约办事模块：预约流程图如下图4.4，用于实现用户预约律师办事，包括查看律师信息，律师分律所、级别搜索，预约，查看预约列表，取消预约等；

关于律所模块：用于查看律所的相关信息，包括律所简介、业务说明、核心团队成员、律所的最新动态；

联系反馈模块：用于介绍各个分所的联系方式，并提供一个可以提交意见建议的用户反馈平台。

4.2.2 后台管理模块设计

用于实现管理员登录后台，修改密码，对数据进行增、删、改、查等操作，从而管理前台的动态数据，本模块设计包括管理员管理、用户管理、律所分所管理、新闻管理、业务范围管理、预约管理、日志管理、常见问题管理、常见问题类型管理、信息搜索等。

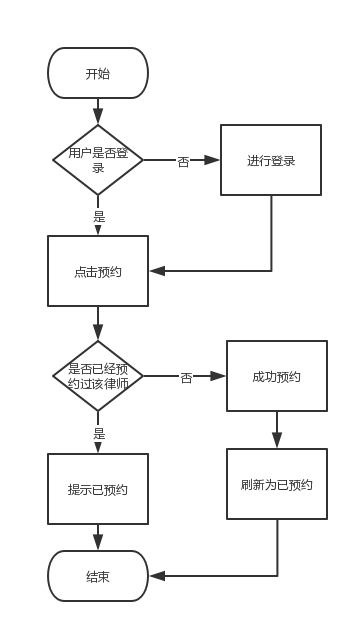


图4.4 预约流程图

4.3 数据库设计

数据库表的总体设计为：admin管理员表，用于存储管理员用户信息；adminlog管理员登录日志表，用于存储管理员登录后台的日志；lawyer律师表，用于存储律师信息；news新闻表，用于存储律所新闻信息；office律所信息表，用于存储分所简介；oplog操作日志，用于后台操作信息；reservation预约表，用于存储预约信息；scope业务范围表，用于存储律所业务信息；suggestion意见建议表，用于存储用户的意见建议信息；user用户表，用于存储用户信息；user\_log用户登录日志表，用于存储用户登录后台的信息；question\_type常见问题类型表，用于存储常见问题类型；question常见问题表，用于存储常见问题。具体设计如下表。

表4.1 管理员表admin

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 其他 |
| id | 编号 | int | NOT NULL AUTO\_INCREMENT |
| name | 管理员账号 | varchar | DEFAULT NULL |
| pwd | 密码 | varchar | DEFAULT NULL |

表4.2 管理员登录日志表adminlog

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 其他 |
| id | 编号 | int | NOT NULL AUTO\_INCREMENT |
| admin\_id | 所属管理员 | int | DEFAULT NULL |
| ip | 登录IP | varchar | DEFAULT NULL |
| addtime | 登录时间 | datetime | DEFAULT NULL |

表4.3 律所信息表office

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 其他 |
| id | 编号 | int | NOT NULL AUTO\_INCREMENT |
| name | 分所名称 | varchar | DEFAULT NULL |
| logo | 标志图片 | varchar | DEFAULT NULL |
| director | 主任 | varchar | DEFAULT NULL |
| addtime | 添加事件 | datetime | DEFAULT NULL |
| introduction | 主任介绍 | text | 一 |

表4.4 新闻表news

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 其他 |
| id | 编号 | int | NOT NULL AUTO\_INCREMENT |
| name | 新闻标题 | varchar | DEFAULT NULL |
| introduction | 新闻内容 | text | 一 |
| addtime | 添加时间 | datetime | DEFAULT NULL |

表4.5 律师表lawyer

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 其他 |
| id | 编号 | int | NOT NULL AUTO\_INCREMENT |
| name | 姓名 | varchar | DEFAULT NULL |
| grade | 级别 | int | DEFAULT NULL |
| logo | 照片 | varchar | DEFAULT NULL |
| introduction | 简介 | text | 一 |
| content | 办案经历 | text | 一 |
| address | 地址 | text | 一 |
| is\_here | 是否在所 | tinyint | DEFAULT NULL |
| is\_recommended | 是否值班 | tinyint | DEFAULT NULL |
| office\_id | 所属分所 | int | DEFAULT NULL |
| addtime | 添加时间 | datetime | DEFAULT NULL |

表4.6 业务范围表scope

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 其他 |
| id | 编号 | int | NOT NULL AUTO\_INCREMENT |
| name | 业务名 | varchar | DEFAULT NULL |
| addtime | 添加时间 | datetime | DEFAULT NULL |
| introduction | 业务介绍 | text | 一 |

表4.7 操作日志表oplog

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 其他 |
| id | 编号 | int | NOT NULL AUTO\_INCREMENT |
| admin\_id | 所属管理员 | int | DEFAULT NULL |
| ip | 操作IP | varchar | DEFAULT NULL |
| reason | 操作原因 | varchar | DEFAULT NULL |
| addtime | 登录时间 | datetime | DEFAULT NULL |

表4.8 预约表reservation

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 其他 |
| id | 编号 | int | NOT NULL AUTO\_INCREMENT |
| user\_id | 用户编号 | int | DEFAULT NULL |
| user\_name | 用户名 | varchar | DEFAULT NULL |
| lawyer\_id | 律师编号 | int | DEFAULT NULL |
| lawyer\_name | 律师名 | varchar | DEFAULT NULL |
| addtime | 添加时间 | datetime | DEFAULT NULL |

表4.9 意见建议表suggestion

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 其他 |
| id | 编号 | int | NOT NULL AUTO\_INCREMENT |
| name | 昵称 | varchar | DEFAULT NULL |
| email | 邮箱 | varchar | DEFAULT NULL |
| content | 意见内容 | text | 一 |
| addtime | 注册时间 | datetime | DEFAULT NULL |

表4.10 常见问题类型表question\_type

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 其他 |
| id | 编号 | int | NOT NULL AUTO\_INCREMENT |
| name | 类型名 | int | DEFAULT NULL |
| addtime | 添加时间 | datetime | DEFAULT NULL |

表4.11 用户表user

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 其他 |
| id | 编号 | int | NOT NULL AUTO\_INCREMENT |
| username | 用户名 | varchar | DEFAULT NULL |
| pwd | 密码 | varchar | DEFAULT NULL |
| email | 邮箱 | varchar | DEFAULT NULL |
| phone | 手机号 | varchar | DEFAULT NULL |
| face | 头像 | varchar | DEFAULT NULL |
| addtime | 注册时间 | datetime | DEFAULT NULL |

表4.12 用户登录日志表user\_log

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 其他 |
| id | 编号 | int | NOT NULL AUTO\_INCREMENT |
| user\_id | 所属用户 | int | DEFAULT NULL |
| ip | 登录IP | varchar | DEFAULT NULL |
| addtime | 登录时间 | datetime | DEFAULT NULL |

表4.13 常见问题类型表question

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 描述 | 类型 | 其他 |
| id | 编号 | int | NOT NULL AUTO\_INCREMENT |
| name | 类型名 | int | DEFAULT NULL |
| addtime | 添加时间 | datetime | DEFAULT NULL |

5 系统实现

5.1 系统首页

系统首页包括搜索律师、值班律师、在所律师等功能。

搜索律师的实现，如图5.1，由于预约办事模块也设置搜索律师，故将search作为公共部分，通过使用include标签实现部分引用。表单含有两个部分，前一半按分所（office.name）查询律师，后一半按级别（lawyer.grade）查询，一共分为三级，初级律师用一星代表，中级律师用二星代表，高级律师用三星代表。



图5.1 律师搜索实现效果图

分页显示实现，使用paginate函数实现分页，使用page表示当前页码，per\_page表示每页显示的数量，分页效果图如图5.2所示。



图5.2 律师搜索分页显示效果图

值班律师的实现，如图5.3，在首页可以查看今日值班律师，实现过程为筛选出is\_recommended为1的律师，作为值班律师，公示在首页，点击图片可以跳转至律师详情页。

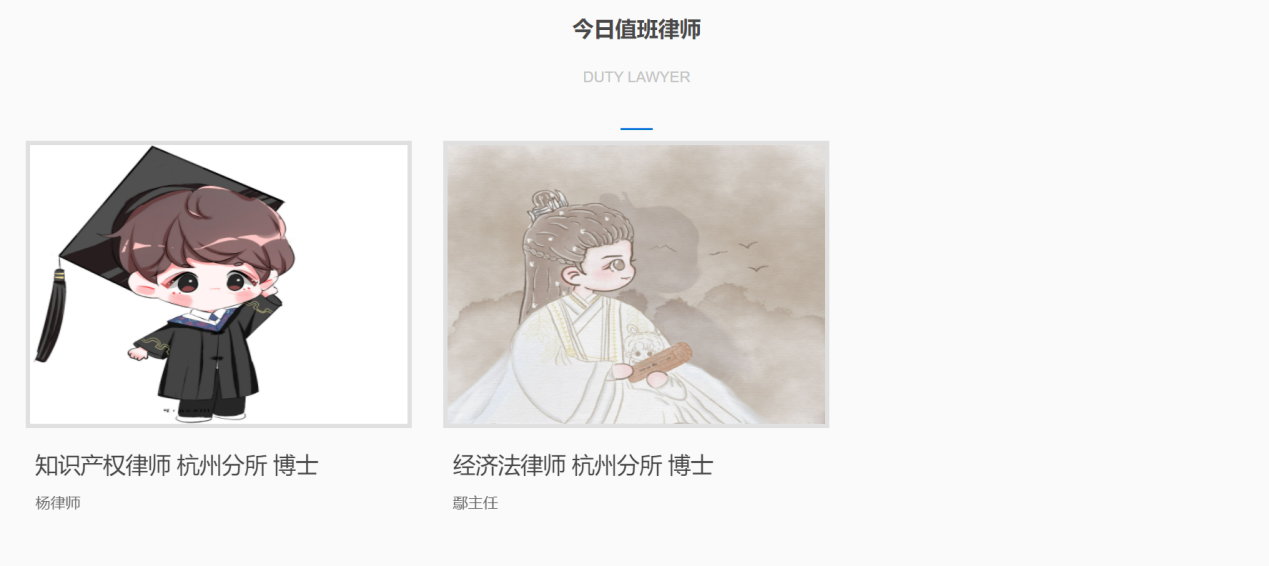


图5.3 值班律师实现效果图

在所律师的实现，如图5.4，首页可以查看没有外出的律师，实现过程为筛选is\_here为1的律师，公示在首页，点击律师图片可以跳转至律师详情页。

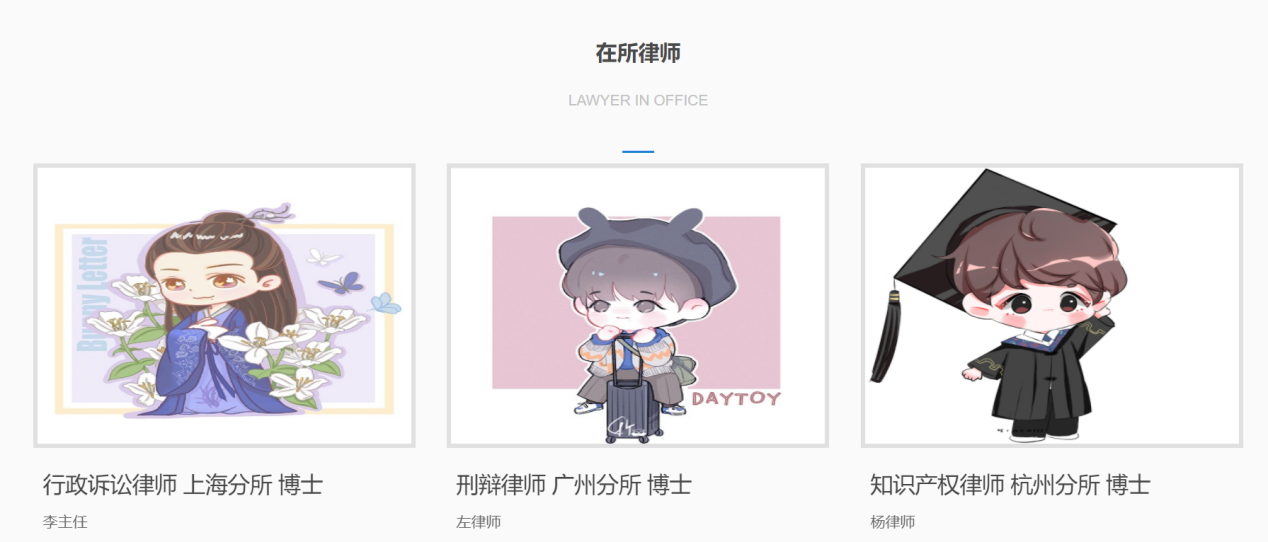


图5.4 在所律师实现效果图

5.2 注册登录模块

注册模块实现，如图5.5，用户使用浏览器访问“/register”，匹配路由，执行register函数。首先实例化RegisterForm类，RegisterForm类继承自FlaskForm，可以实现表单的创建和验证。接下来，执行views.py文件中的register函数，使用render\_template渲染模板register.html，并传递form变量，form.csrf\_token生成一个隐藏字段，有跨站请求伪造（Cross-Site Request Forgery，CSRF）令牌避免CSRF攻击。验证验证码填写正确，提交表单后，验证表单，用户名、邮箱、密码不得为空且符合格式，即两次密码相同且邮箱唯一，然后存入数据库。



图5.5 注册页面实现效果图

登录模块实现，如图5.6，登录模块和注册模块业务逻辑类似，首先实例化LoginForm表单，以GET方式访问路由，则执行渲染页面的功能，POST方式访问，则填写登录信息登录，验证表单，若信息错误，提示修正，信息正确，则将用户信息写入Session。



图5.6 登录页面实现效果图

退出时，使用session.pop()清空Session中的user\_id，再重定向至登录页面即可。

5.3 事务咨询模块

智能咨询模块实现，如图5.7，可以在下拉框中先选择常见问题类型，再选择常见问题，进而达到便捷输入的效果，也可以在输入框中输入问题，实现个性化输入。智能咨询的实现过程为利用爬虫抓取网站用户问题和律师回复，并通过NLP技术转换为数据集，导入项目，运行decode\_conv.py，将原始数据转换为对话（ask和answer的键值对）形式的数据，然后保存成sqlite3数据库格式，在db文件夹下生成conversation.db数据库文件；运行data\_preprocess.py，在db文件夹下生成dictionary.json字典文件；运行data\_utils.py，在db文件夹下生成conversation.txt文件以及在bucket\_dbs文件夹下面生成四个对应不同对话长度的桶数据文件；运行s2s.py，模型训练，生成log日志文件夹和model模型参数文件夹；运行sever.py，构建模型，初始化变量，渲染页面，最后调用回复，显示结果，实现一问一答。

模型实现分为四步：初始化、训练、处理batch和预测。1）初始化：定义一个损失函数，使用sampled\_softmax\_loss，定义前向操作返回embedding\_attention\_seq2seq，定义编码器输入输出、loss损失函数计算的权重系数，开始构建模型：测试阶段，使用model\_with\_buckets获取输出和损失值，每个桶分别设置loss的可视化；训练阶段，计算梯度和参数更新：获取所有训练参数，定义优化器。最后定义模型持久化对象，以便维护。2）训练：根据桶id获取当前桶对应的训练对象、梯度norm值、loss值、获取最后一个时刻的decoder输出，添加所有时刻对应的输出。3）处理batch：获取bucket\_id相应桶的ask和answer的字符长度大小限制值，将汉字转化为可识别的id，将长度调节为一致，即Encoder和Decoder Padding，再分批Encoder，Decoder。4）预测中先判断参数位于哪个桶中，再处理输入句子，给出编码解码输入，执行模型，最后获取输出值。

模型训练：先确定学习率、批量梯度下降的批量大小、LSTM每层神经元数量、LSTM的层数、训练轮次等，再建立模型，而后进行训练。训练过程为：获取数据，数据预处理、开始建模、初始化变量和模型恢复，计算每个桶的累计占比，然后开始训练：随机选择一个要训练的桶，从随机的桶中获取数据，进行迭代训练。最后再进行一次模型持久化输出。



图5.7 智能咨询效果图

人工咨询模块实现，如图5.8，与咨询类似，将回复设置为人工回复即可实现连续问答。



图5.8 人工咨询效果图

5.4 预约办事模块

查看律师功能实现，预约页面可以分页查看所有律师。实现过程为首先使用get\_or\_404方法根据ID判断律师是否存在，如果不存在直接跳转404页面，如果存在，使用session.get获取用户的ID，根据用户ID和律师ID判断是否已经预约该律师，使用|safe正常显示数据库中律师办案经历的格式到HTML页面，实现的核心代码如下。

$(document).ready(function () {  
 $(".collect-button").click(function () { // 触发点击事件  
 user\_id = { user\_id }; // 获取用户ID  
 if(!user\_id){ // 如果用户ID不存在，即用户未登录  
 layer.msg("请先登录",{icon:2,time:2000}); //弹出错误信息  
 return false; // 终止执行  
 }

预约律师功能实现，律师详情页（info）可以点击预约按钮，实现预约，首先判断用户是否登录，如果已登录，则通过Ajax异步提交方式实现预约律师功能，实现的核心代码如下，效果图如图5.9。

var lawyer\_id = {lawyer.id}; // 获取ID  
$.ajax({ // 使用Ajax异步提交  
 url: "{{ url\_for('home.reservation\_add') }}", // 提交到的URL  
 type: "GET", // 提交方式为GET  
 data:{lawyer\_id: lawyer\_id}, // 传递参数  
 dataType: "json", // 数据类型为json  
 success: function (res) { // 操作成功后执行逻辑  
 if (res.ok == 1) {  
 layer.msg("预约成功，请于下一个工作日的工作时间来律所办事！",{icon:1,time:2000}); // 显示弹出层信息  
 $(".collect-button").empty(); // 清空按钮区文字  
 $(".collect-button").append("已预约"); // 填充文字  
 } else {  
 layer.msg("您已预约",{icon:2,time:2000}); // 提示  
 }



图5.9 律师预约实现效果图

取消预约功能实现，我的预约（reservation\_list）页面，可以点击取消预约，取消预约与预约类似，使用Ajax异步提交，当用户单击取消预约按钮时，获取律师ID，提交至reservation\_cancel，使用session.get函数获取当前用户ID，然后查询reservation表，最后删除数据，返回json数据，核心代码如下。效果图如图5.10。

id = request.args.get(**"id"**, **""**) *# 获取律师D*user\_id = session[**"user\_id"**] # 获取当前用户ID  
reservation = Reservation.query.filter\_by(id=id, user\_id=user\_id).first()   
if reservation: # 如果存在  
 db.session.delete(reservation) # 删除数据  
 db.session.commit() # 提交数据  
 data = dict(ok=1) # 写入字典  
else:  
 data = dict(ok=-1) # 写入字典  
import json # 引入json模块  
return json.dumps(data) # 输出json格式

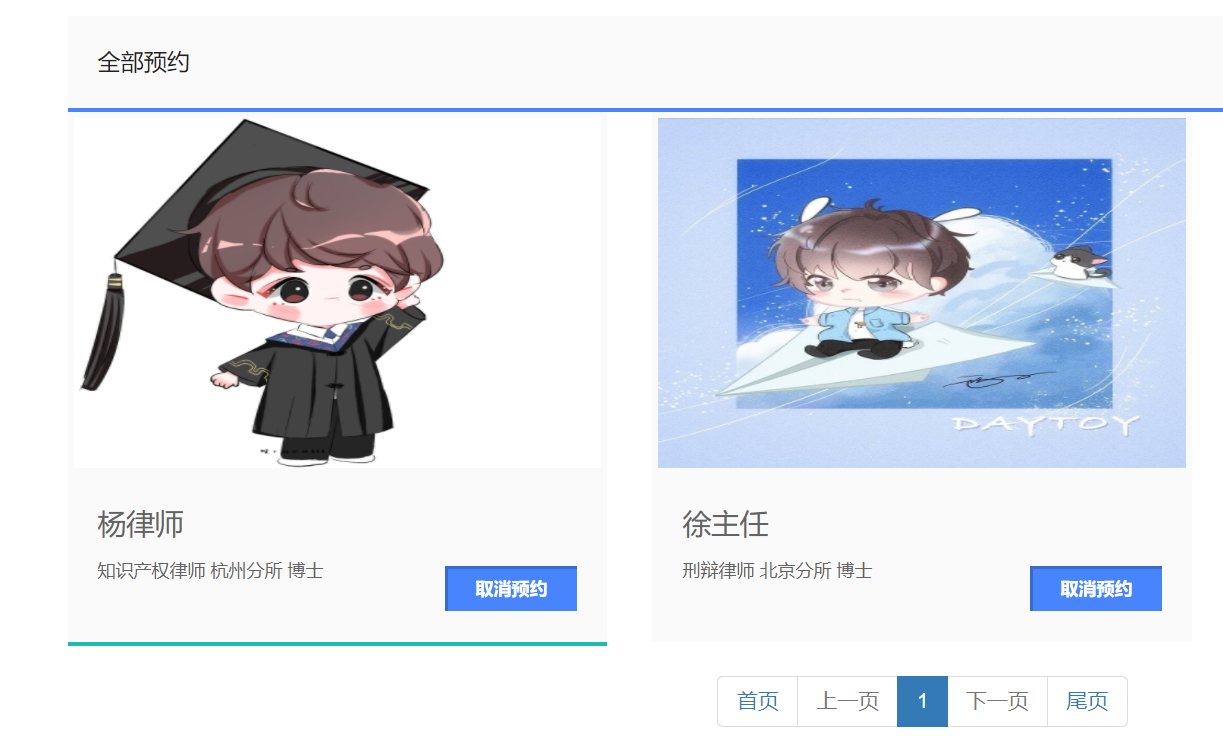


图5.10 取消预约实现效果图

5.5 关于律所模块

关于律所模块实现，点击关于我们，下方页面显示分为四个部分。

关于蓝海，即律所简介，为静态页面，如图5.11。



图5.11 律所简介实现效果图

团队成员，为北京、上海、广州、杭州四个分所的图片（office.logo）和文字说明（office.name），以及各个分所的律所主任（office.director）和简介（office.introduction），如图5.12。

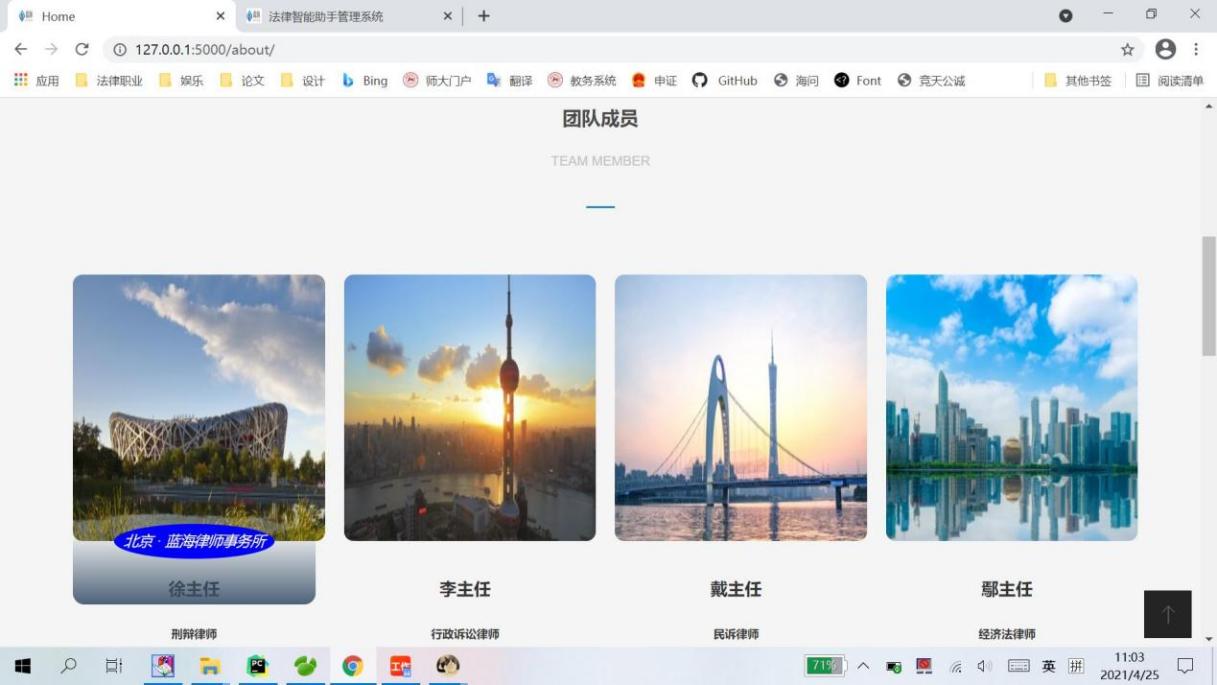


图5.12 团队成员实现效果图

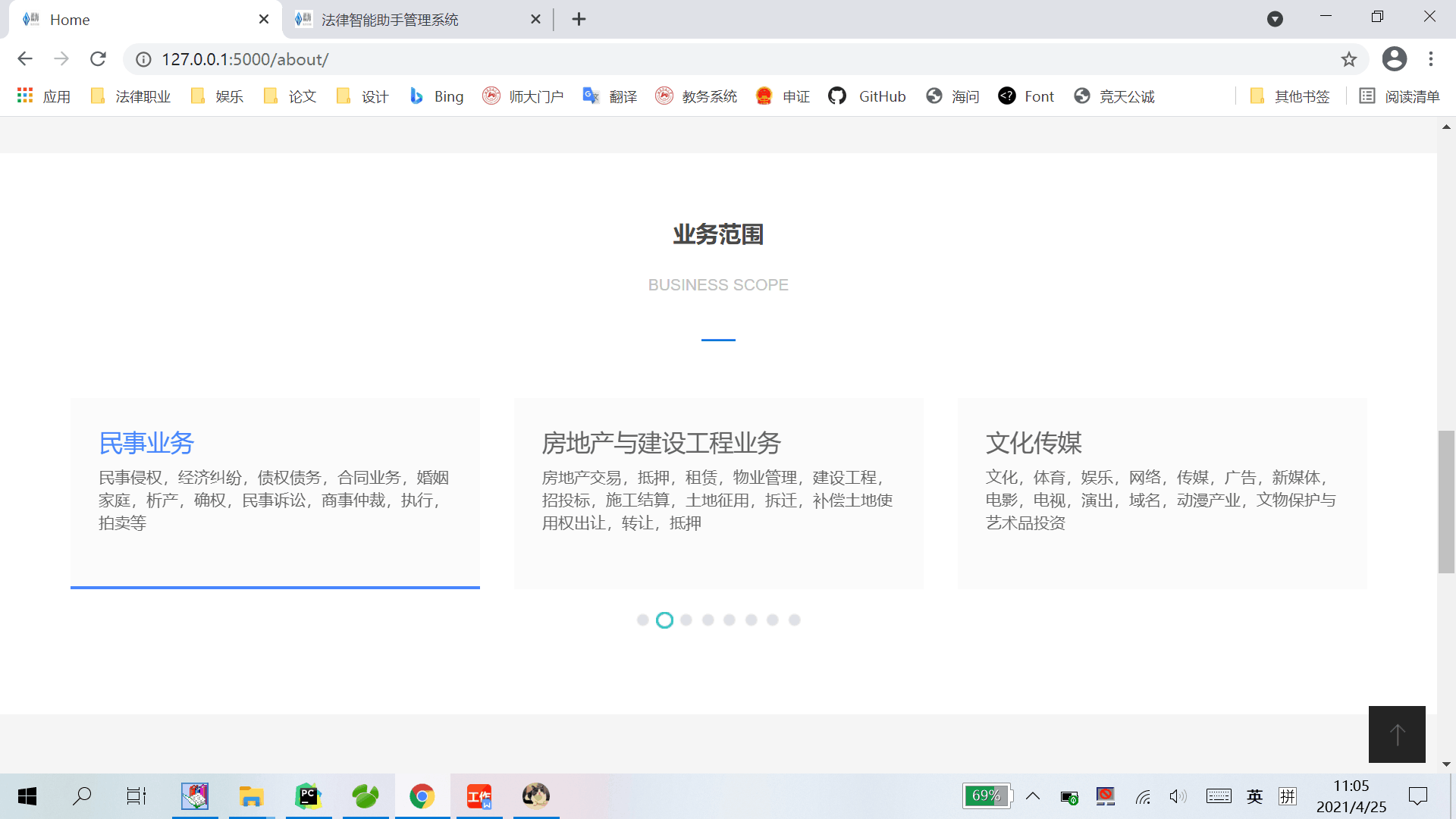
业务范围，点击下方轮播按钮，可以查看律所的业务名称（scope.name）以及业务内容（scope.introduction），如图5.13。

图5.13 业务范围实现效果图

新闻资讯，可以查看律所的新闻标题（news.name）、新闻内容（new.introduction），如图5.14。

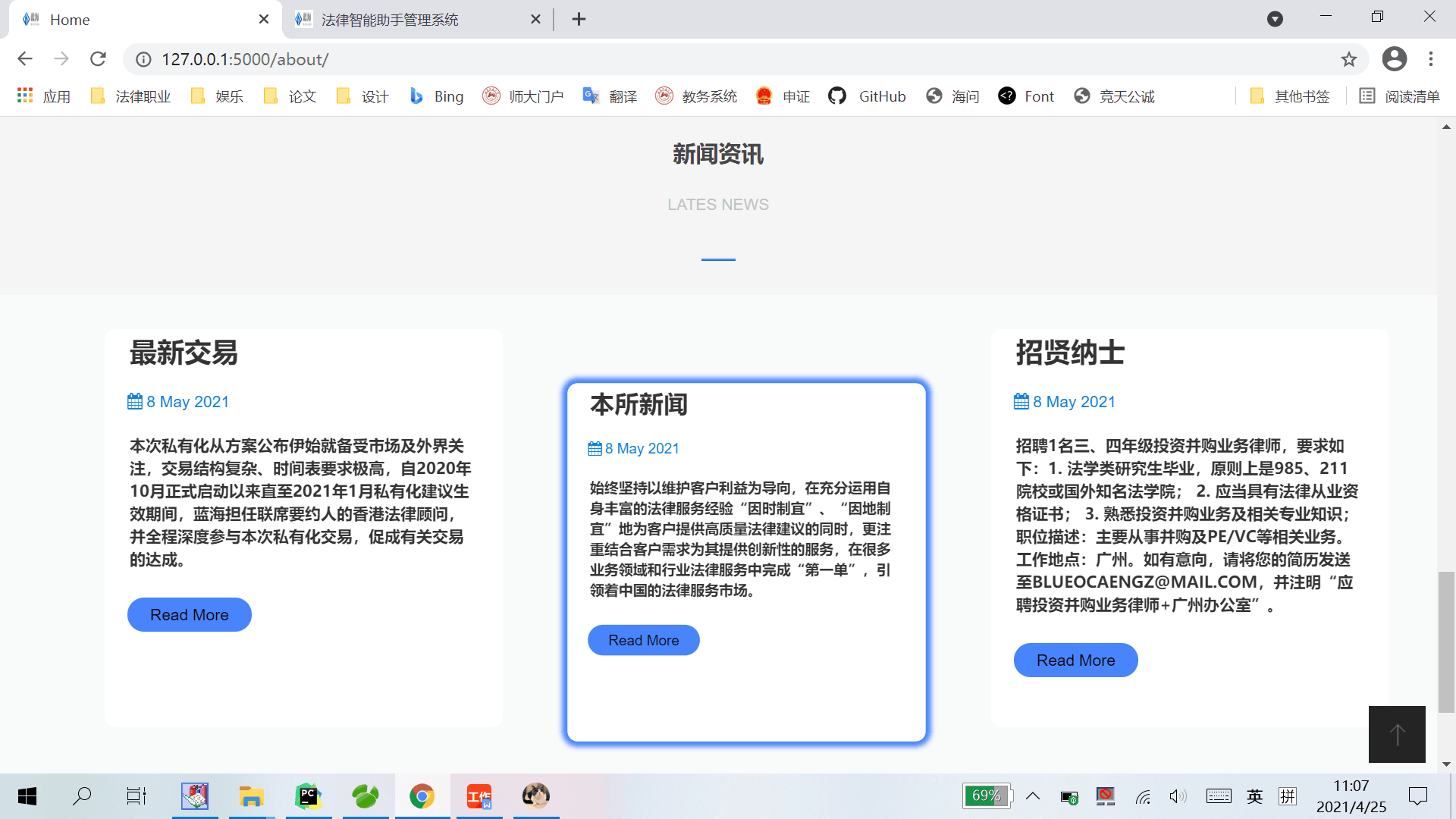


图5.14 新闻咨询实现效果图

5.6 联系反馈模块

联系与反馈主要分为两部分，左边为联系我们，以静态页面为主，如图5.15。



图5.15 联系我们实现图

意见建议实现与登录注册功能类似，当以GET方式访问时，只做页面渲染，当用户单击发送消息时，以POST方式访问，首先在SuggestionForm类中进行表单检测，姓名、邮箱、意见内容不能为空，错误则显示提示信息，验证通过则写进数据库，实现效果如图5.16。



图5.16 意见反馈实现效果图

5.7 后台管理模块

管理员登录实现，如图5.17，在后台登录页面/admin，填写管理员账号和密码，单击登录，即可实现管理员登录。若输入账号不存在，密码为空，密码不正确，则显示提示信息，若输入正确登录成功，则进入首页，显示控制面板。



图5.17 管理员登录实现效果图

控制面板实现，如图5.18，左侧为一个菜单，显示所有管理模块，访问每个模块均需要验证管理员登录与否，故定义一个admin\_login来判断是否登录，session有值则为登录，为空则跳转至登录页面；右侧为律所简介和特色模块，主要为静态页面。



图5.18 后台控制面板实现效果图

由于管理模块实现相似，本文篇幅有限，故仅以律师管理模块为代表展开介绍。

律师管理模块实现，包括新增律师、律师列表、编辑律师、删除律师等功能。

新增律师实现，如图5.19。首先实例化表单，使用WTForms扩展实现表单验证在form.py中设置验证规则，律师名不能为空，所属律所为下拉列表，默认选择第一个北京分所，律所所在地不能为空，级别下拉列表默认为3星，单选是否值班，是否在所，上传律师封面不能为空，且需要设置FileField和FileAllowed方法，确定允许上传的文件类型为jpg或png，律师简介不能为空，律师介绍不能为空，验证通过则执行添加律师的业务逻辑，首先查看是否有重复添加，再单独处理上传律师封面，保证文件的安全性，核心代码如下。再为律师的各个属性赋值，而后添加并提交数据，写入日志，最后进行页面跳转，渲染模板。

file\_logo = secure\_filename(form.logo.data.filename) # 使文件名唯一  
if not os.path.exists(current\_app.config[**"UP\_DIR"**]): # 创建多级目录  
 os.makedirs(current\_app.config[**"UP\_DIR"**]) # 创建文件夹  
 os.chmod(current\_app.config[**"UP\_DIR"**], **"rw"**) # 设置权限  
logo = change\_filename(file\_logo) # 更改名称  
form.logo.data.save(current\_app.config[**"UP\_DIR"**] + logo) # 保存

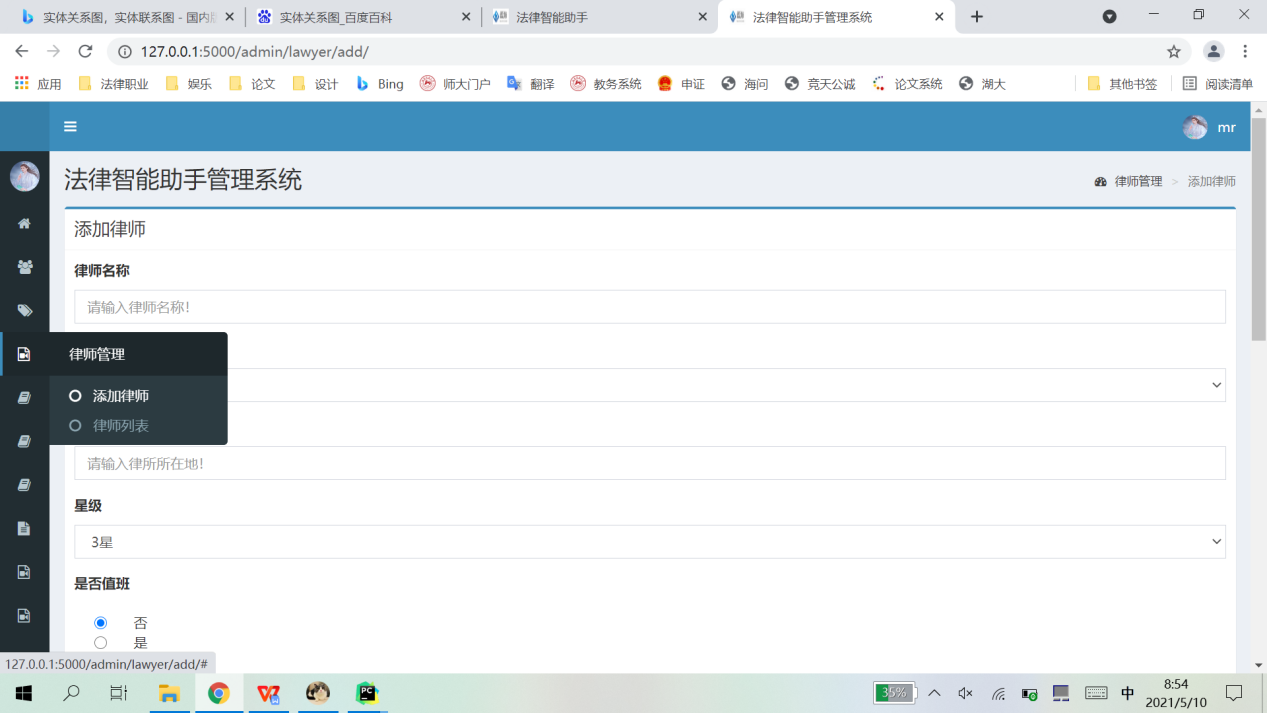


图5.19 添加律师实现效果图

律师列表实现，如图5.20，添加完律师后可以点击律师列表查看已添加的律师列表，律师按照添加时间的倒序方式显示。



图5.20 律师列表实现效果图

在律师列表页面也可以使用右上角的搜索框，搜索律师名字查找目标律师，如图5.21。核心代码如下。

name = request.args.get(**'name'**,**''**,type=str) # 获取查询标题  
page = request.args.get(**'page'**, 1, type=int) # 获取page参数值  
if name : # 根据标题搜索  
 page\_data = Lawyer.query.filter(Lawyer.name.like(**"%"** + name + **"%"**)).order\_by(  
 Lawyer.addtime.desc() # 根据添加时间降序  
 ).paginate(page=page, per\_page=5) # 分页  
else : # 显示全部  
 page\_data = Lawyer.query.order\_by(  
 Lawyer.addtime.desc() # 根据添加时间降序  
 ).paginate(page=page, per\_page=5) # 分页  
return render\_template(**"admin/lawyer\_list.html"**, page\_data=page\_data)



图5.21 搜索律师实现效果图

编辑律师实现，如图5.22，在律师列表中单击编辑律师，则可进入编辑页面，编辑律师页面与添加页面基本相似，只是在渲染模板时，要传递当前ID的律师数据。



图5.22 编辑律师实现效果图

删除律师实现，如图5.23，当律师出现人事变动时，可以在律师列表页面单击删除按钮，删除律师，核心代码如下。

lawyer = Lawyer.query.get\_or\_404(id) # 根据律师ID查找数据  
db.session.delete(lawyer) # 删除数据  
db.session.commit() # 提交数据  
flash(**"律师删除成功"**, **"ok"**) # 使用flash存储成功信息  
addOplog(**"删除律师"**+lawyer.name) # 添加日志  
return redirect(url\_for(**'admin.lawyer\_list'**, page=1)) # 渲染模板



图5.23 删除律师实现效果图

6 系统测试

6.1 功能测试

功能测试最常用黑盒测试，即主要考虑数据的输入输出，不考虑复杂的中间逻辑[23]。本系统功能测试结果如表6.1。

表6.1 功能测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 标题 | 测试条件 | 操作步骤 | 期望结果 | 测试结果 |
| 注册登录 | register/login模块完成，form表单无误 | 运行程序，进入注册登录模块，空输入 | 提示输入不能为空 | √ |
| 错误的用户名或密码或验证码 | 提示用户名不存在或密码错误或验证码错误 | √ |
| 输入正确的用户名、密码 | 成功登录 | √ |
| 事务咨询 | 数据集导入并训练完成，advisory模块完成 | 智能咨询，能够自动生成较为合理的答案 | 一问一答 | √ |
| 人工咨询，能够得到回复 | 连续问答 | √ |
| 预约办事 | reservation模块完成 | 搜索查看律师 | 分页显示搜索结果 | √ |
| reservation模块完成，用户未登录 | 点击预约律师 | 提示先登录 | √ |
| reservation模块完成，用户已登录 | 预约律师已经预约过，点击预约 | 提示已预约 | √ |
| 预约的律师尚未预约，点击预约 | 提示预约成功，加到预约列表 | √ |
| 在预约列表中点击取消预约 | 提示取消预约并刷新页面 | √ |
| 联系反馈 | contact模块完成，form表单无误 | 输入信息为空 | 提示输入信息不能为空 | √ |
| 输入邮箱格式不正确 | 提示输入格式错误 | √ |
| 输入正确 | 提示成功提交 | √ |
| 后台管理 | admin模块完成，form表单无误 | 管理员账号密码输入为空 | 提示输入不能为空 | √ |
| 管理员账号或密码错误 | 提示账号不存在或密码错误 | √ |
| 添加、编辑管理事务输入为空 | 提示输入不能为空 | √ |
| 添加、编辑管理事务上传文件格式不正确 | 提示上传文件格式不正确 | √ |
| 删除管理事务 | 删除后，刷新列表 | √ |

6.2 页面测试

本系统页面简洁大方，显示正常，并无乱码、多余字符等问题，对特殊的部分测试如表6.2。

表6.2 页面测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 标题 | 测试页面 | 操作步骤 | 期望结果 | 测试结果 |
| 首页 | index | 点击导航栏各项 | 跳转到相应功能页面 | √ |
| 服务、咨询、团队的介绍 | 轮播文字 | √ |
| 搜索框 | 显示分律所，分级别筛选 | √ |
| 值班律师 | 筛选出is\_recommend为1的律师显示 | √ |
| 在所律师 | 筛选出is\_here为1的律师显示 | √ |
| 注册  登录 | register | 点击登录 | 跳转至登录页面 | √ |
| 点击验证码 | 刷新出一个新的验证码 | √ |
| login | 点击注册 | 跳转至注册页面先行注册 | √ |
| 咨询 | chart | 点击人工咨询 | 可以在智能咨询页面点击跳转至人工咨询页面 | √ |
| 预约  办事 | reservation | 搜索查看律师 | 分页显示搜索到的律师 | √ |
| 点击在所律师 | 跳转至在所律师 | √ |
| 预约律师已经预约过，点击预约 | 显示为已预约，点击后提示您已预约 | √ |
| 预约的律师尚未预约，点击预约 | 显示为预约，点击后显示已预约 | √ |
| 在预约列表中点击取消预约 | 刷新为减少取消预约后的页面 | √ |
| 关于  我们 | about | 查看律所简介 | 显示律所信息 | √ |
| 查看团队成员 | 显示分所和主任及其介绍 | √ |
| 查看业务范围 | 显示业务名称和业务内容 | √ |
| 查看最新资讯 | 显示新闻标题，及新闻内容 | √ |
| 联系  反馈 | contact | 查看分所联系方式 | 显示各个分所联系方式，标志图片 | √ |
| suggstion | 建议表单 | 显示建议提交表单及提交按钮 | √ |
| 后台  管理 | admin/login | 登录 | 显示登录表单及提交按钮 | √ |
| admin | 点击菜单控制面板 | 显示律所介绍和特色模块 | √ |
| 点击菜单事务添加 | 显示可以编辑的添加页面 | √ |
| 点击菜单事务列表 | 显示搜索框，事务编辑按钮和删除按钮 | √ |
| mr | 点击头像 | 显示退出或修改密码 | √ |

6.3 性能测试

对智能咨询模块的训练结果，进行采样测试，测试结果如下表。

表6.3 原句测试

|  |  |  |
| --- | --- | --- |
| 问题 | 训练轮次 | 训练结果 |
| 车辆转让协议书对方没有履行怎么办 | 70 |  |
| 53 |  |
| 35 |  |

表6.4 删除部分信息测试

|  |  |  |
| --- | --- | --- |
| 问题 | 训练轮次 | 训练结果 |
| 转让协议书对方没有履行怎么办 | 70 |  |
| 53 |  |
| 35 |  |

表6.5 调换语序测试

|  |  |  |
| --- | --- | --- |
| 问题 | 训练轮次 | 训练结果 |
| 对方没有履行车辆转让协议书怎么办 | 70 |  |
| 53 |  |
| 35 |  |

训练轮次为70，仅能回答出与原数据集相同的回答，说明存在一定的过拟合现象，训练轮次为35，对调换语序的问题回答表现略差，最终的训练轮次取35与53中间的44轮次。

结论

本文首先讨论了法律智能助手的研究背景，研究现状，介绍了用到的技术，结合实际情况进行需求分析，完成系统设计、数据库设计，功能模块设计，然后步入系统实现。本系统按照功能可以分为登陆注册模块、咨询模块、预约办事模块、查看律所信息模块、联系与反馈模块、以及后台管理模块。

利用爬虫机制抓取网站用户提问和律师回复，NLP技术处理数据形成数据集（问答对），利用SQLite存储数据集，采用Seq2Seq模型结合Attention机制训练数据集完成智能咨询的自动回复。通过Flask框架实现值班律师、注册登陆、预约办事、关于律所、联系反馈、后台管理等功能模块。

最后还进行了系统测试，通过对系统进行的功能测试、页面测试、性能测试可得系统基本满足使用需求，对提高律所的品牌知名度、推进律所信息化、系统化、智能化提供了支持和保障。

虽然律师事务所行政管理智能助手系统的功能基本得以实现，但因设计开发时间相对较短，同时受自身的水平、精力等条件的制约，系统仍然有需要优化之处。

1）随着时间推移，用户量、数据量必将日渐增加，因此，有必要改进算法、整合系统，加快其响应速度；

2）智能咨询的数据集筛选受到时间、精力和法学知识水平等各方面条件制约，仍有需要改进之处；

3）算法仍有进一步的完善空间，桶的大小，训练的轮次亦可再经斟酌。

参考文献

[1]许良光．律师事务所管理系统的设计与实现[D]．长春：吉林大学，2015．

[2]卢盛宽，罗承芙．律师事务所的行政管理工作[J]．中国律师，2014，2014(10)：46-46．

[3]KATHRANIP．Artificial intelligence and the future of law[J]．Artificial Intelligence & Law，2017，10(10)：78-84．

[4]黄俏娟，罗旭东．人工智能与法律结合的现状及发展趋势[J]．计算机科学，2018，45(12)：1-11．

[5]张妮，杨遂全，蒲亦非．国外人工智能与法律研究进展述评[J]．法律方法，2014，16(002)：458-480．

[6]陈文山．基于柔性工作流的律师事务所管理系统设计与实现[D]．上海：上海交通大学，2008．

[7]林嘉荣．基于WEB的律师事务管理信息系统设计和实现[D]．成都：电子科技大学，2012．

[8]周倚文．面向法律领域的问答系统研究[D]．长沙：湖南大学，2018．

[9]吴永大．浅议法律咨询系统功能需求分析[J]．法制博览，2019，2019(24)：66-68．

[10]石达，芦天亮，杜彦辉．法律助手APP的设计与实现[J]．网络安全技术与应用，2021，2021(01)：62-63．

[11]黄薇屹．基于知识图谱的深度法律内容问答模型[D]．深圳：中国科学院大学(中国科学院深圳先进技术研究院)，2020．

[12]李武波，张蕾，舒鑫．基于Seq2Seq的生成式自动问答系统应用与研究[J]．现代计算机(专业版)，2017，2017(36)：57-60．

[13]吴威震．基于seq2seq模型的聊天机器人对话研究[D]．南京：南京邮电大学，2019．

[14]Marie-Francine Moens，Peter Spyns，Paulo Quaresma，Irene Pimenta Rodrigues．A Question Answer System for Legal Information Retrieval[J]．Frontiers in Artificial Intelligence and Applications，2005，2005(12)：134．

[15]董晓娟．金马律师事务所信息化体系的构建与实施[D]．哈尔滨：哈尔滨工业大学，2016．

[16]刘路．基于序列到序列模型的答案生成方法研究[D]．武汉：武汉科技大学，2019．

[17]涂海，彭敦陆，陈章，等．S2SA-BiLSTM面向法律纠纷智能问答系统的深度学习模型[J]．小型微型计算机系统，2019，40(05)：1034-1039．

[18]俞鸿飞．序列到序列模型优化的若干方法研究[D]．苏州：苏州大学，2020．

[19]董叶豪，柯宗武，熊旭辉，等．基于Seq2Seq的问答系统应用研究[J]．福建电脑，2018，34(10)：122-123．

[20]吴威震．基于seq2seq模型的聊天机器人对话研究[D]．南京：南京邮电大学，2019．

[21]洪源．基于深度学习的智能问答系统设计[D]．镇江：江苏科技大学，2018．

[22]简治平，蔡爱雄，李欣，等．基于Seq2Seq模型的问答系统研究[J]．广东石油化工学院学报，2019，2019(6)：48-52．

[23]田成林．基于B/S模式的律所综合管理信息系统的设计与实现[D]．沈阳：东北大学，2016．