上海交通大学硕士学位论文

基于法律知识推理对科创板上市审核结果的预测

硕士研究生：马振文

学　　号：118037930075

导　　师：李国强副教授

申请学位：专业学位硕士

学　　科：软件工程专业

所在单位：电子信息与电气工程学院

答辩日期：2021 年 1 月 6 日

授予学位单位：上海交通大学

Dissertation Submitted to Shanghai Jiao Tong University
for the Degree of Master

PREDICTION OF THE RESULTS OF THE INITIAL PUBLIC OFFERING REVIEW ON THE SCIENCE AND TECHNOLOGY INNOVATION BOARD BASED ON LEGAL KNOWLEDGE REASONING

Candidate:	Zhenwen Ma
Student ID:	118037930075
Supervisor:	Prof. Guoqiang Li
Academic Degree Applied for:	Master of Engineering
Speciality:	Software Engineering
Affiliation:	School of Electronic Information and Electrical Engineering
Date of Defence:	January 6, 2021
Degree-Conferring-Institution:	Shanghai Jiao Tong University

基于法律知识推理对科创板上市审核结果的预测

摘　要

法律是服务于社会经济发展的一大利器。法律非诉业务中首次公开募股（俗称，上市）常常是一项耗时耗力的巨大项目。借助于科技的发展，尤其是最近大火的人工智能（AI）产业的飞速发展，法律也逐渐走向科技化，信息化的道路。科创板注册制的实行使得上市的信息更加公开，透明，同时也为预测上市审核的结果提供了可能。知识图谱被认为是最具希望成为具有认知能力的人工智能的关键技术，其高密集的知识形态正好符合上市所需的大量法律知识。如何通过科技的手段预测科创板上市审核的结果，其关键也就在于如何构建一个可靠的法律知识图谱，如何利用法律知识图谱来推理预测。因此构建一个可靠的符合法律推理逻辑的法律知识图谱，是预测科创板上市审核结果的必经之路。本文的主要研究成果包括:

（1）设计了一个可推理的法律知识库的知识表示 schema。本文针对知识的结构和法律的本体论进行分析，从科创板首发上市领域里的可靠的法律文本知识源中，对法律知识进行了文本建模，设计了一套可推理的法律知识 schema。

（2）搭建了一个抽取嵌套知识结构的标记平台并利用其进行知识抽取。基于本文提出的嵌套知识结构的表示，本文对嵌套知识结构的标记平台进行了需求分析，设计了总体架构和其中主要的可视化组件，并给出了详细的实现方法。并对法律知识进行了抽取，设计了法律知识的存储方式，进而利用此标记平台构建了基于嵌套结构的法律知识图谱数据集。

（3）转化了法律知识图谱中的法律规则并判断出了法律规则结果。对科创板首发上市的法律知识进行应用架构设计，应用了获得的法律知识库。其中对法律知识进行了法律规则转换，形成了法律道义逻辑

公式作为法律推理大前提。应用爬虫收集器提取了在科创板上市的事件信息，获得推理所需的小前提。最终通过判决器获得了法律规则结果。

（4）对科创板首发上市审核结果进行了预测。介绍了科创板上市的相关规则，设计了相关的预测策略。基于判决器生成的法律规则结果集，采用了人工智能的 SVM 算法作为本文的推理内核。实验结果表明，科创板首发上市审核结果预测 F1 值近 80%，说明本文的提取的法律知识是有用的，本文的法律知识应用框架是有效的。

关键词：知识图谱，道义逻辑，法律规则，首发上市，知识推理

PREDICTION OF THE RESULTS OF THE INITIAL PUBLIC OFFERING REVIEW ON THE SCIENCE AND TECHNOLOGY INNOVATION BOARD BASED ON LEGAL KNOWLEDGE REASONING

ABSTRACT

Law is a great tool for social and economic development. In the legal non-litigation business, the initial public offering (be called for short as IPO) is often a huge time-consuming and labor-intensive project. With the help of the development of science and technology, especially the rapid development of the artificial intelligence (AI) industry in recent fires, the law has gradually moved towards technological and informatization. The implementation of the registration system on the Science and Technology Innovation Board makes the IPO information more open and transparent, and it also makes it possible to predict the outcome of the IPO review. Knowledge graph is considered to be the most promising key technology of artificial intelligence with cognitive ability, and its high-density knowledge form is just in line with the large amount of legal knowledge required for IPO. How to predict the results of the STAR Market IPO review through scientific and technological means lies in how to construct a reliable legal knowledge graph and how to use the legal knowledge graph to reason and predict. Therefore, building a reliable legal knowledge graph that conforms to the logic of legal reasoning is the only way to predict the results of the STAR Market IPO review. The main research results of this article include:

(1) Design a knowledge representation schema for a reasonable legal knowledge base. This paper analyzes the structure of knowledge and the

ontology of law. From the reliable legal text knowledge sources in the field of IPO on the Science and Technology Innovation Board, the text models the legal knowledge and designs a set of reasonable legal knowledge schema.

(2) A marking platform for extracting nested knowledge structures was built and used for knowledge extraction. Based on the representation of the nested knowledge structure proposed in this article, this article analyzes the requirements of the marking platform of the nested knowledge structure, designs the overall architecture and the main visual components, and gives a detailed implementation method. The legal knowledge was extracted and the storage method of legal knowledge was designed, and then the legal knowledge graph data set based on the nested structure was constructed using this marking platform.

(3) The legal rules in the legal knowledge graph are transformed and the result of the legal rules is judged. Design the application framework of the legal knowledge for the IPO on the Science and Technology Innovation Board, and apply the acquired legal knowledge base. Among them, the legal knowledge was transformed into legal rules, and the legal moral logic formula was formed as the major premise of legal reasoning. The application crawler collector extracts the event information listed on the Science and Technology Innovation Board, and obtains the small premises required for reasoning. Finally, the legal rule result was obtained through the Judger.

(4) The results of the IPO review on the Science and Technology Innovation Board were predicted. Introduced the relevant rules for IPO on the Science and Technology Innovation Board and designed relevant forecasting strategies. Based on the result set of legal rules generated by the Judger, the artificial intelligence SVM algorithm is used as the reasoning core of this article. The experimental results show that the predicted F1 value of the IPO review on the Science and Technology Innovation Board is nearly 80%, indicating that the legal knowledge extracted in this article is useful and the

legal knowledge application framework in this article is effective.

KEY WORDS: Knowledge graph, moral logic, legal rules, Initial Public Offering, knowledge reasoning

目 录

插图索引

表格索引

算法索引

第一章　绪论

1.1　研究背景及意义

中国特色社会主义的建设离不开依法治国。法律涉及生活中的各个角落，在国家大力推行依法治国的今天，如何让法律更公平更高效地运作，是一个十分热门的研究领域。比如在计算法学领域，如何让法律用可计量的方式服务于社会，让法律像数学一样精确，是一个普遍研究的话题。随着智能时代的来临，人工智能高科技技术的大力发展，势必会对法律行业产生质的飞跃。其中，知识图谱是人工智能技术的重要发展方向之一，俗话说："你给我多少知识，我给你多少智能"[1]。如"上海刑事案件智能辅助办案系统"，基于大量刑事案件，对刑事案件的罪名，刑期和相关的法律条文做预测，极大地提高了法官、检察官和公安部门的效率。但是法律是需要被人信服的，基于大量案件统计出来的结果，又如何能实现个案公平？于是基于可解释性的研究，是法律人工智能的一个重要目的。

首次公开募股（Initial Public Offering）英文缩写为 IPO，俗称上市，是法律非诉业务中的一个较大的版块。科创板（The Science and Technology Innovation Board; 简称为 STAR Market）是最先实现的注册制上市，是国家为了促进高新科技产业的发展，在上海证券交易所下新设的一个版块。科创板设计的规则众多，涉及法律《中华人民共和国公司法》，《中华人民共和国证券法》以及部门法规《上海证券交易所科创板股票发行上市审核规则》，《上海证券交易所科创板股票发行上市审核问答》等等众多的法律法规。基于我国是一个成文法国家，法律规则主要来源就是基于人民代表大会制定的法律法规，或国务院制定的行政法规、部门规章等法律。有法有据，是人民、社会、国家运作基本模式。并且在依法治国的方针指导下，法律会越制定越细，规则会越来越多，社会将更加井井有条。一旦矛盾出现，可以迅速地找到对应的法律条文。但是有些规则之间交叉复杂，若非一些在专业领域长期从事法律的人士，也很难弄清楚其中的法律问题。

法律知识图谱是对法律知识的整合，积累，是走向法律人工智能的必经之路。知识的粒度越细，我们对知识的理解越深入。基于知识就可以很好地对个案提出专有的解决方案。就如同人类智能，是在不断地获取知识，增长经验。如何构建一个法律知识图谱，让法律更加智能，让法律会思维，是我们的研究方向。人工智能走向强人工智能的过程中肯定要对知识能够理解。知识图谱就是对现实世界中的各种知识进行提取整合。本项目就是对科创板上市领域的法律知识图谱的构建进行探索研究，并依据该图谱推理科创板上市的结果。

1.2 国内外研究现状

法律智能是近几年新兴的领域，很多研究者和企业进入到该领域，企图对法律业进行科技化，智能化。法律领域也有很大的发展空间，因为在中国，人均拥有的律师比例远低于其他国家。法律又深入生活的各个方面。无论是从需求上还是从效率上都需要更多的研究者参与进来。

无论是法律专家，还是科技工作人员都需要共同努力才能使法律更加智能。

1.2.1 国外研究现状

国外的研究有很多，COLIEE (Competition on Legal Information Extraction and Entailment) 是外国的一项赛事，它有四项任务：法律案例检索，法律案例确定任务，法规法检索任务，法律问答任务[2]。这四项任务都需要各种法律知识和计算机自然语言理解的知识，来相互补充，对法律信息进行处理，进而归纳为一个分类或排序的问题。

Branting 这篇文章对于 WIPO 的域名争议问题，分析了目标案件的特征，制定了 schema 用 The MITRE Annotation Toolkit 自动标记[3]。Tolan 的研究发现国家或地区，性别，年龄等要素特征来分析预测犯罪会导致很大程度的不公平[4]。Doesburg 提出了 Calculemus 方法，去构建一个形式化的框架 Formal Language for the INTerpretation of sources of norms (FLINT)，这些框架是行为框架，义务框架和事实框架，并针对 Catholic Marriage 案例进行了研究[5]。这些研究都表明，每一个法律领域，它所涉及的问题都需要更加高效、智能的法律科技，来帮助他们更好的完成任务。

Prakken 介绍了很多逻辑的法律应用，特别关注法律论证的逻辑模型，发现逻辑的现代法律应用证实了从推论到信息流，论证和互动的逻辑范围不断扩大的最新趋势[6]。Domingos 将逻辑和概率进行了巧妙的结合，提出了马尔科夫逻辑网[7]。Li 基于马尔科夫逻辑网预测了离婚案件的法律判决[8]。El. 开发了一种有根据的法律本体，用于基于规则的推理，为此，提出了一种中间，协作和模块化的方法，或者已重用基础本体和核心本体以简化本体的开发，在基于同类本体的方法中采用生成的本体，以使用逻辑语言 SWRL 形式化刑法法律规则列表[9]。

1.2.2 国内研究现状

国内研究刚刚起步，2018 年 CAIL 中国法研杯开始举办，目前已经在举办第 3 届了。同时上海 206 工程，“上海刑事案件智能辅助办案系统”也顺利完成。该系统以上千万篇法律文书，法律条文等法律文件为数据，融合深度学习模型，和

专家经验，对刑事案件进行分析，已初见成果。但是其主要任务，法律判决预测，却只能对简单案件进行预测，属于端到端的黑盒模型。而法律需要的是公正，而不是机器直接给出一个结果。机器给出的结果如果没有可解释性，那它的参考价值将大大降低。TopJudge 提出了一个深度学习模型，它将法律判决预测中的子任务，进行分析发现法律条文可以推出罪名预测，法律条文和罪名预测又可以对刑期预测提供帮助，这三个子任务之间的拓扑关系依次给与深度学习模型取得了比较好的效果[10]。

相似案件匹配这一法律任务也是由来已久，律师，法官，甚至普通的老百姓，他们都可以根据自己的描述来找到相似的案件，以此来提供参考。从另一个方面讲，相似案件匹配是一种语义匹配，基于自然语言处理技术 TF-IDF，BM25 等是传统的统计方法。更有许多深度学习的方法，比如 Luo 提出了一种基于注意力的神经网络框架，在中国刑事案件判决文件上的实验结果表明，模型在指控预测和相关文章提取方面均有效[11]。

法律问答也是一项很有意义的任务，人们对律师的咨询就可以看成一种法律问答任务。只不过法律概念晦涩难懂，对于机器也很难理解，目前已有的自然语言处理技术如 NLU，就很好的针对该任务。但是比起人类的效果还是相差甚远。同时今年 CAIL 又开了一个比赛通道，数据集是司法考试的客观题[12]。机器预测的结果只有不到人类的一半，而一个没有经过训练的人，他的表现也只有 60 % 多一点。可见对于法律这一专业领域还是有很多难题需要攻克。CJRC 数据集[13] 来自中国判决文件，而问题由法律专家注释，可以帮助研究人员通过阅读理解技术来提取元素。

IFlyLegal 是一个综合系统，可通过利用深层上下文表示和各种注意机制等技术来执行法律咨询，多方法律搜索和法律文件分析。IFlyLegal 是第一个采用最新 NLP 技术并能够满足不同用户群体（例如律师，法官，检察官和客户）需求的中国法律系统[14]。

法律智能目前主要有两种研究方法，一种是基于向量化的方法，另一种是基于符号的方法。向量化的方法，比如传统的 Word2vec[15]，和预训练模型，比如 Bert[16]，目的在于设计高效的神经网络来实现高准确的表现，甚至有人发布了 Legal-Bert[17]。Ilias[18] 提出了 BERT 的分层版本，它绕过了 BERT 的长度限制，法律判断预测了欧洲人权法院的案件。而符号的方法，基于可理解的手工绘制的符号来做法律任务。Verheij[19] 将案例、规则、参数三者之间的紧密形式联系在一起得到了进一步发展，以试图表明案例可以为建立领域中的规则和论点提供逻辑依据。更精确的说，符号方法更聚焦于用可解释的法律知识在文档中进行推理，像

法律事件，法律关系等；同时向量化的方法尝试学习到大数据中现有的特征来预测。这两种方法显然，可解释的符号方法并不是很高效，而基于向量化的方法有更好的表现但是却不具有可解释性。因此，现实当中这些现存的方法很难应用起来。

无论是向量化的方法还是符号的方法，它们的难点在于 3 个：1，法律知识的建模，法律有很多领域，每个领域的概念各不相同，而法律文本确是有较好格式化的。2，法律推理，法律的推理和其他 NLP 中的推理不一样，法律推理需要更精确更严格。将预定义的规则融入法律智能当中十分必要的。除此之外，更复杂案件场景需要更加复杂的推理模式。3，可解释性。法律判决，应该是能被人民所理解的，能够呈现出公平正义的。可解释性和模型的表现是同等重要的。

知识图谱与关系推理是一个交叉领域的研究，推理是将已知的事实经演绎，归纳，溯因，预测，推断和类比得出新知识的过程。根据知识图谱我们可以推理出新的事实，新的关系，新的规则等。它是由实体、关系和特定的图谱结构形成的一个图网络。基于知识图谱，可以进行本体概念推理等。很多实验表明带有知识的预训练模型比普通的预训练模型效果要好。而将向量化方法和符号方法相结合起来也是一个理所当然的方向。法律知识对于两种方法都起到不可或缺的作用，如何获得法律知识，并将法律知识转化成计算机可理解的信息，是非常重要的。

关于科创板上市注册制，文献 [20] 分析了其中的法律逻辑，指出了科创板注册制改革的价值追求和预期目标。文献 [21] 分析了科创板注册制下交易所发行上市审核权能的配置。

1.3 本文主要工作

本研究的核心是构建可靠的符合法律推理逻辑的法律知识图谱，并探索法律知识图谱在科创板首发上市审核的应用。基于对法律理论的学习，通过对法律知识图谱的构建，发现法律规则之间内在的逻辑，应用到具体案例时准确给出问题所在的推理过程，让法律人工智能的推理更具可信性，可解释性，以达到法律所要求的权威性。基于科创板首发上市审核的预测，可以表明知识图谱所具有的庞大知识体系，知识推理能力，以及对复杂法律问题的解决能力。如图1–1为基于法律知识推理的科创板上市预测系统架构图。本研究的主要工作对应图中模块可以分为以下几个部分：

一、本体层：可信赖可推理的法律知识库的本体设计。针对法律本体论的研究与分析。我们寻找了一些可靠的知识源。然后从科创板上市领域的法律条文，部门规章，行业规范等不同知识源获取所需文本，通过对文本的学习抽象认识，对

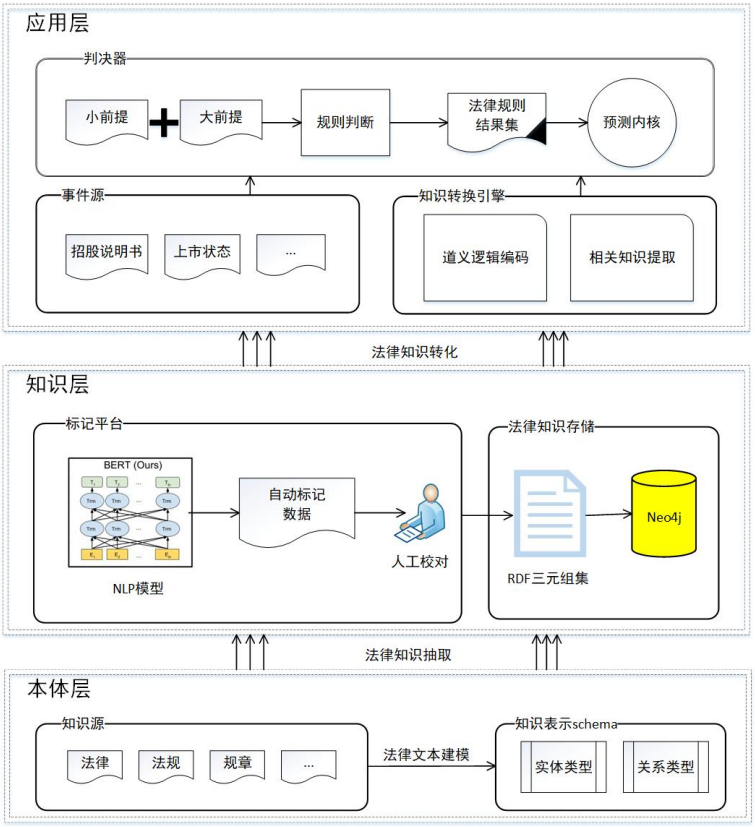

图 1–1 基于法律知识推理的科创板上市预测系统框架图

Figure 1–1 The framework of IPO prediction in the STAR Market based on legal knowledge reasoning

法律文本进行建模。最后设计出一种嵌套知识结构的法律知识表示 schema，其中包括实体类型和关系类型等本体概念。

二、知识层：通过嵌套知识结构的文本标记平台的构建来对法律知识进行抽取。首先我们分析了嵌套知识结构的文本特征需求，然后设计了标记平台的系统架构和实现方式。基于此平台，我们可以标记更加复杂的知识结构，使知识的表达能力更强。最后我们将标记好的法律知识存放在图数据库中。

三、应用层：对法律知识图谱的法律规则转化及应用。我们将充分利用法律知识库，对其进行知识转换，法律知识编码，法律知识应用。其实是模拟法律推理过程，基于大前提和小前提推理出科创板上市审核结果的推理模式。

四、对科创板上市审核结果的预测结果与实验。我们对科创板首次公开募股进行业务分析与研究，制定详细的预测策略，利用人工智能算法预测法律知识库的推理结果，对法律知识库进行具体的应用和实际的验证。

1.4 论文结构

本文一共包含七章，其具体内容安排如下：

第一章绪论。对本文内容进行研究背景介绍，研究意义分析，并介绍了国内外研究现状。综合的表述了本文的具体做法，以及文章结构安排。

第二章相关技术简介。本章对文中用到相关技术进行介绍，以及构建知识图谱所具有的挑战，现有的相关解决方案。包括知识图谱技术及法律道义逻辑，中文自然语言处理技术、存储技术、Web 技术等。

第三章法律知识图谱的 schema 设计。本章主要介绍了法律知识图谱的 schema 设计过程，就是对法律知识进行建模，即获取科创板上市领域的可靠知识源、制定法律知识表示 schema，包括实体类型和关系类型、法律知识关系约束。

第四章嵌套知识结构标记平台的构建并对利用其对知识进行抽取，知识存储。本章通过对具有嵌套知识结构的文本进行分析，进而构建出具有应用价值的嵌套知识结构标记平台。具体包括需求分析、架构设计、可视化组件设计以及标记平台的前端模块、标记组件模块和后端模块的实现。之后我们利用构建好的标记平台对知识源进行知识抽取，最后将知识存放进图数据库中。

第五章法律知识图谱的法律规则转化及应用。本章设计了法律知识应用的架构。通过对法律道义逻辑的分析，根据法律推理三段论，对法律知识进行转换并编码为法律道义逻辑公式作为大前提，通过爬虫收集器获得小前提，最后通过判决器获得法律规则结果集。判决器根据此结果预测上市的应用架构。

第六章对科创板首发上市审核结果的预测。本章通过对科创板首发上市的分析，设计了相关的预测策略。基于判决器生成的法律规则结果集，我们采用了人工智能的 SVM 算法作为我们的推理内核。最后我们展示了实验结果，表明法律知识应用框架是有效的。

第七章总结与展望。本章回顾了全文的内容，先是总结了本文的研究及其研究成果。再是对项目的不足之处指出了一些未来的工作方向。

第二章　相关技术简介

本章会先介绍知识图谱的相关概念，以及法律知识图谱在本文中的具体应用。随后介绍了本文推理所使用的法律道义逻辑。然后介绍如何构建知识图谱的自然语言处理技术，最后介绍了在搭建构建平台时用到的 Web 技术，标记平台为本文的知识抽取提供了强大的支持。

2.1　知识图谱的概念及应用

2.1.1　知识图谱的概念

知识图谱这个词最早被用在了 1972 年的这个论文里[22]，不过现代知识图谱的兴起是在 2012 年 Goolge 提出了 Goolge Knowledge Graph 的概念[23]。它的目的是提升搜索引擎的准确度，实践表明已经取得了非常不错的效果。

知识图谱的应用十分广泛，近年来人工智能技术又飞速发展，知识图谱又出现了前所未有的发展势头。知识图谱在智能问答，搜索引擎，对话系统等技术领域得到飞速发展。

知识图谱可以积累和表达知识，现实世界中，人类的思考也是基于知识的，因此人类也模仿出类人思考的知识结构，表达一个知识点和另一个知识点两者之间的关系。知识图谱符合基于图的数据模型，该模型可以是有向边标记图，属性图等[24]。

知识图谱确实具有很强的知识表达能力。具体表现为，它可以由一种三元组（头部实体，关系，尾部实体）和（实体，属性，属性值）组成的描述自然世界中物体与物体，以及物体本身属性的有向图结构。形式上表现就是由 subject->predicate->object 组成的有向图结构。

知识表示是研究采用怎样的表示方法被计算机计算模拟的。目前常见的知识表示方法有万维网，语义网，以及 OWL[25] 和 RDF[26]。如图 2–1为知识表示图。其中 URI/IRI 是资源的网络链接，XML 和 RDF 都是一种资源的表示方法。SPARQL 是一种较为常用的知识查询语言。右侧蓝色区域覆盖的部分是知识推理模块常常用到的，其中 RDFS 和 OWL 是可以推理的知识表示方法。最上层就是 Trust 和 Interaction 的部分。

知识图谱的构建方法很多，其中包括人工构建，和自动化构建。人工构建是指由领域专家，将知识奉献出来，对领域里的知识进行总结，构建成一个知识图

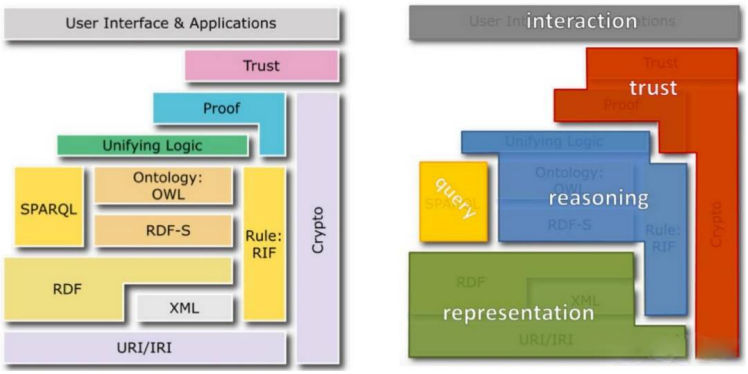

图 2–1 知识表示

Figure 2–1 Knowledge represent

谱。自动化构建是指利用自动化技术和工具如自然语言处理技术构建成一个知识图谱。也有分为自底向上和自顶向下的方法。

知识图谱具有可观的推理能力。如果将经典逻辑中的推理方式，集合进知识图谱中将获得不可低估的推理能力。比如，演绎推理，归纳推理，溯因推理，类比推理等等推理方法是知识推理的主要方法。推理的具体形式可以表现在关系上，也可以表现在实体的属性上。知识图谱是一个充满信息的知识库。由于其具有推理能力而获得较好的可解释性。

但是知识图谱也有缺点。知识图的缺点主要是由“双重一阶谓词逻辑”作为知识表示的固有缺陷引起的。长期以来，研究人员一直在不懈地追求和探索知识表示，如（头部实体，关系，尾部实体）之类的表示方法，可以代表大多数简单事件或实体属性，但对于复杂知识无能为力。

不过本文对复杂知识进行了分析，提出了一种基于嵌套关系的知识图谱表示框架。所谓嵌套关系是指（头部实体，关系，尾部实体）三元组中的头部实体或尾部实体也可能是一个三元组。有了这样一种结构，知识图谱的表达能力更强了，推理能力也得到了很大的提升。比如天变黑导致下雨和变凉，有两个三元组（天变黑，导致，下雨）和（天变黑，导致，变凉），但是不能直观的看到这种并列关系，而一个嵌套三元组（天变黑，导致，（下雨，和，变凉））就可以很直观表示这种并列关系。当然其中的关系可以根据需要，随意定义。这样知识图谱的表达能力就上升了一个更高的台阶。

2.1.2 知识图谱在本课题中的应用

本文主要应用和构建法律知识图谱，鉴于法律所涉及的领域众多，本文主要致力于科创板首次公开募股的法律知识图谱建设和应用。

法律知识图谱在法律人工智能（LegalAI）中扮演着十分重要的角色。法律本身要求很强的逻辑性，可解释性都可以在知识图谱上得到体现。法律有其自身的道义逻辑。道义逻辑简单来说就是要求人们应该怎么做，可以怎么做，和哪些事不能怎么做。通过知识图谱的可视化展示能很清楚的表现出来法律这一条条网络。这也是诸多法学家希望看到的景象。

尽管法律领域的知识图谱方法很有前途，但在实际使用之前仍存在两个主要挑战。首先，LegalAI 中知识图谱的构建很复杂。在大多数情况下，没有现成的法律知识图谱可用，因此研究人员需要从头开始。此外，在不同国家的法律体系下，不同的法律概念具有不同的表示和含义，这也使构建一般的法律知识图谱具有挑战性。一些研究人员试图嵌入法律词典，这可以被视为替代方法。其次，广义法律知识图谱的形式与 NLP 中常用的形式不同。现有的知识图谱表达的是实体与概念之间的关系，但是 LegalAI 更加关注法律概念的解释。这两个挑战使得通过嵌入 LegalAI 的知识建模变得不容易，研究人员可以尝试克服未来的挑战[12]。

本课题将构建一个科创板的法律知识图谱，该图谱将利用我们新提出的嵌套知识图谱的概念，虽说构建的过程很复杂，但是我们争取更多的应用自动化的方法。首先，我们寻找可靠的法律知识源，比如现行的法律法规，而不是通过案例里面的内容来作为我们的知识源；其次，我们和法律专家商讨总结出一套法律内容的 schema；再者，我们运用一些自然语言处理技术作为我们知识提取的方法；最后，再和法律专家一起对我们的知识图谱进行知识的纠错和修正形成一份可靠的法律知识图谱。最终，我们将法律知识图谱转化为法律规则，之后再用法律图谱和法律规则进行推理。

2.2 法律道义逻辑

道义“deontic”这个词最早来源于希腊语“δεόυτως”，被翻译为“应有的”或“适当的”。实际上，它是对行为的可解释性进行规范性表达的词语或句子。一些词语约束人们的行为，如“义务”，“职责”，“许可”等等表达词。这些词被称为道义词，句子中含有道义词的被称为道义句。它的出现和宗教、国家等权力机构密切相关。法律道义逻辑常常也被称为义务逻辑或规范逻辑[27]。

法律道义逻辑为本文的 schema 设计提供了理论指导，结合法律道义逻辑提出的相关行为逻辑，本文设计相应的推理方式。本文选用道义逻辑中的较为经典的

等价式来方便机器做推理[28]，例如：

（1）(p→q)→(Op→Oq)

（2）(p→q)→(Pp→Pq)

（3）¬O¬p←→Pp

（4）¬P¬p←→Op

（5）¬Op←→P¬p

（6）¬Pp←→O¬p

（7）Op←→F¬p

（8）Fp←→O¬p

在上述等价式中，O 表示“应该”，P 表示“允许”，F 表示“禁止”。

2.3 自然语言处理技术

世界上的知识，信息大多数都是由自然语言来表示和表达的，自然语言就是我们平时口中所说的语言，而自然语言处理技术就是处理我们所说语言的技术。随着信息化的爆炸，多是自动化的技术。

2.3.1 中文分词

中文分词就是将一段中文语句进行词语分割，类似于英文这种空格分割的语言，中文也需要分割成一个词一个词的句子才能够被计算机作进一步处理。现阶段分词工具已经很成熟，并且也很高效。分词技术的方法主要有基于词典的分词方法和基于统计的分词方法这两种方法[29]。

常用的中文分词工具有：

1. jieba 分词[30]
2. snowNLP
3. PkuSeg[31]
4. thulac[29]
6. StanfoldNLP 分词[32]

2.3.2 实体识别

实体识别（Named Entity Recognition，简称 NER）[33]，具体就是识别出句子中包含某种特殊含义的词语。比如代表人员的词，代表位置的词，代表组织机构的词。这个任务通常包含两个子任务：实体边界识别和实体类型识别。实践中，实

体识别是计算机自然语言处理技术的一项很关键的基本任务。关系抽取大多就是在实体识别的任务之后进行的。

实践中，实体识别的方法有基于隐马尔科夫模型的，基于 BLSTM 模型的，基于注意力机制的。在法律领域多有针对法律文书进行实体识别的，这些研究提升了司法文本的理解能力。BiLSTM-CRF 的实体识别模型在司法领域就取得了比较好的实验结果[34]。如下图2–2所示为司法领域的实体识别的 BiLSTM-CRF 模型。

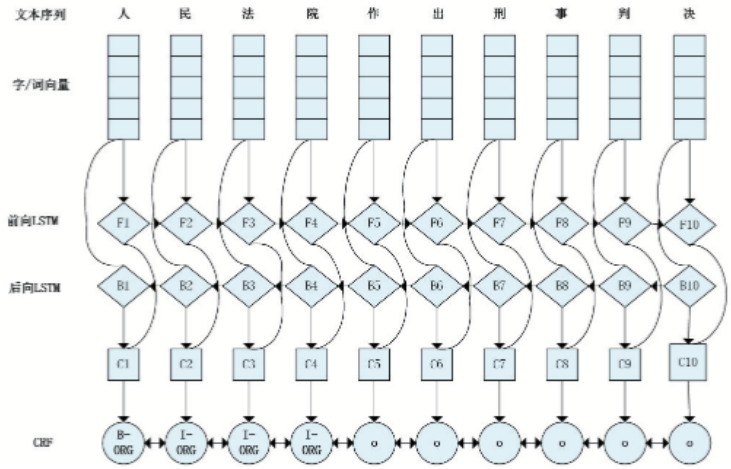

图 2–2 司法实体识别模型

Figure 2–2 Legal NER model

2.3.3 关系抽取

关系抽取（Relation Extraction, 简称 RE）[35] 是构建知识图谱的必不可少的一项技术。关系抽取一般要抽取出句子中表示关系的三元组，如上海交通大学位于上海市 ==> (上海交通大学, 位于, 上海市)。

常见的关系抽取的方法有：有监督方法、半监督方法和无监督方法。基于有无确定的关系集合可以分类为：限定式关系抽取的方法和开放式关系抽取的方法[35]。在实践当中，我们采用的是限定关系抽取，因为我们定义了法律关系的 schema。继而尝试了很多有监督、半监督和无监督的方法，以及它们在嵌套知识结构的关系抽取上的表现。但是任务难度颇高，需要以后详细攻略。

2.4 存储技术

2.4.1 Neo4j

Neo4j 是一个图数据库，它很高效，查询快速，文档丰富，不仅支持实体，还支持关系，不仅有快速的查询效果，还支持多种多样的图数据库查询语言。如图2–3为法律知识库的 Neo4j 查询界面。

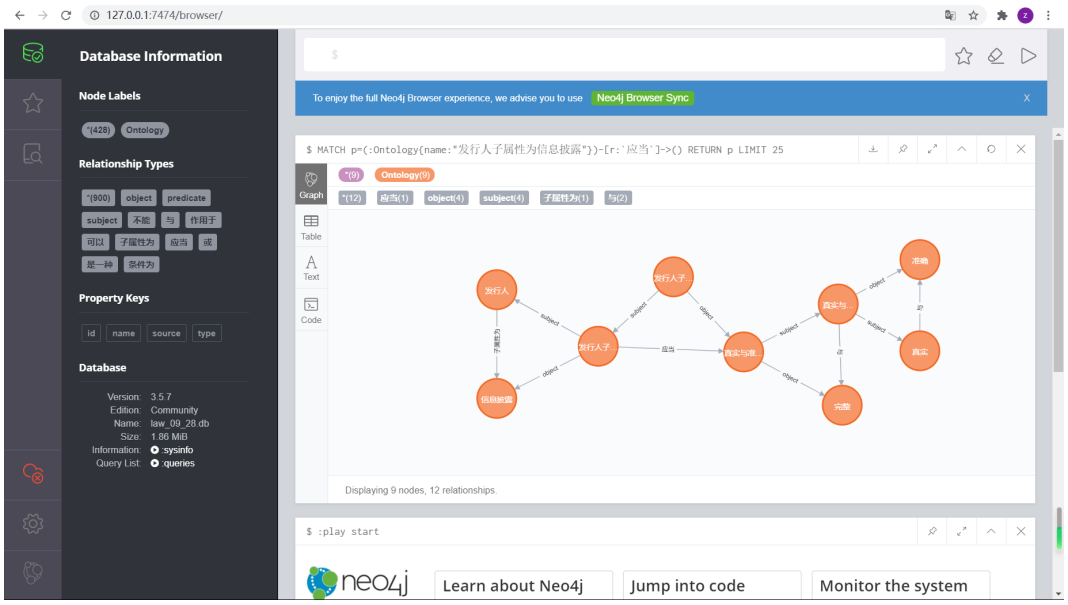

图 2–3 Neo4j 查询界面

Figure 2–3 Neo4j query interface

2.4.2 Elasticsearch

Elasticsearch 是基于 json 格式文档的数据库，支持全文搜索的功能，高效且快速，深受大企业的喜爱。存储方便，高效，还有很多附属的工具可以用。

“ELK” ①是 Elasticsearch、Logstash 和 Kibana 的首字母缩写。Elasticsearch 是一个集搜索和分析引擎的数据库。Logstash 是数据处理管道，亦可转换数据，采集数据，再将数据发送到 Elasticsearch 数据库中。Kibana 用可视化的方式即图表和图形的方式让用户对 Elasticsearch 进行数据操作。如图2–4为项目收集的招股说明书的 Kibana 内容管理视图。

① https://www.elastic.co/cn/what-is/elk-stack

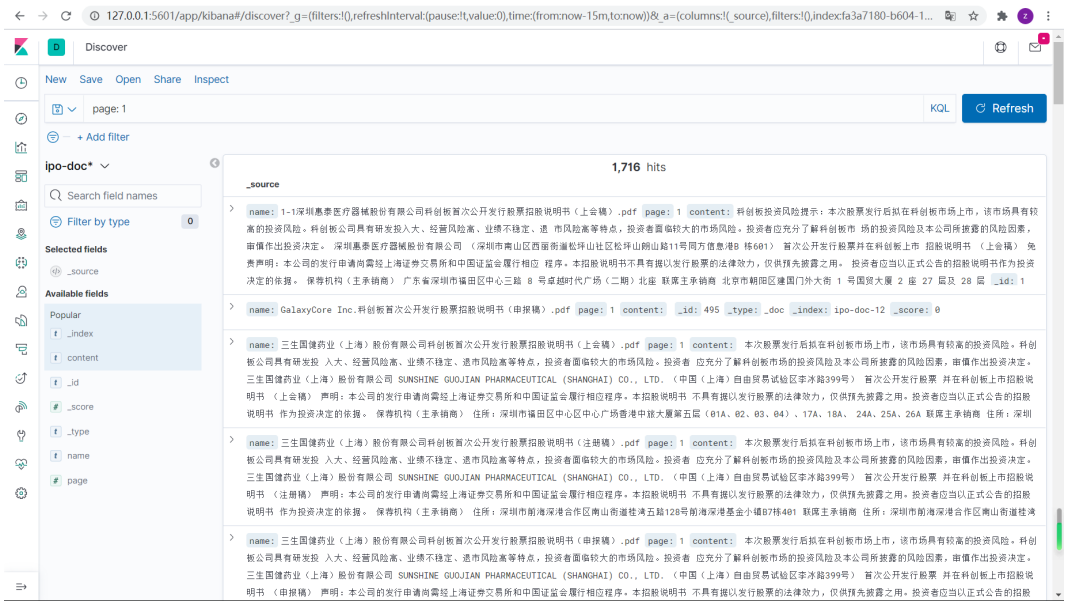

图 2–4 Kibana 内容管理视图

Figure 2–4 Kibana content management view

2.5 Web 技术

Web 技术是信息化技术中网络和互联网技术领域的主要技术。由于每一次用户的信息交流都需要涉及客户端服务和服务器端服务，Web 技术通常可以分为 Web 前端技术和 Web 后端技术。

2.5.1 Web 前端技术

Web 前端技术一般有 HTML,CSS 和 js 等技术栈组成。本课题主要搭建了一个嵌套知识结构的标记平台，用来对复杂的知识结构进行标记的完整的 Web 应用。这里主要应用到前端的一个主流 js 框架，React，它可以提高开发者的效率。

React 使创建 UI 变得不再那么复杂。为应用程序的每种状态设计一个简洁的视图。当数据翻新时，React 可以不仅有效地刷新，还能精确显示组件。React 将 UI 分为娇小的组件级别，并遵循软件工程的单一功能的原则。使用单向数据流的思想来建立数据点，可以从上到下或从下到上构建应用程序。

SVG.js 是一个前端的可视化组件工具，用于操作和动画 SVG 的轻量级库。SVG.js 能够使用 JavaScript 来操作游览器的文档对象，因此使图形具有交互能力。本课题使用 SVG.js 来绘制和操作文档结构，标记实体和关系等。

2.5.2 Web 后端技术

Web 后端是运行在服务器上与数据库相关联的技术，数据存储，数据处理，业务逻辑大部分都在 Web 后端实现。常用的 Web 后端技术有 JAVA 和 Python。Web 后端通过 Https 协议，支持 Restful API 与前端进行交互。

Restful API 是一种互联网用户通信协议，Web 技术应用中广泛应用到它。REST 全称是 Representational State Transfer，中文意思是表述性状态转移。它首次出现在 2000 年 Roy Fielding[36] 的博士论文中，Roy Fielding 是 HTTP 规范的主要编写者之一。Restful API 是一个功能强大，符合软件工程原则的架构。本课题使用 Java 中简单的 https 服务和 Restful API 与前端进行交互。

后端也常常是操作数据库的模块。比如 Mysql，Elasticsearch 和 neo4j 都是后端进行操作的。一般是进行会话来实施一些创建、更新、删除等操作。

2.6 本章小结

本章介绍了知识图谱的概念，以及构建法律知识图谱用到的相关技术及在本课题中的应用。其中法律道义逻辑是设计法律知识图谱 schema 的关键理论指导。自然语言处理技术是构建法律知识图谱的关键技术，包括中文分词，命名实体识别，和关系抽取。存储技术是知识库必备的技术，neo4j 是一个高效的图数据库，elasticsearch 是一个支持全文搜索的数据库。构建嵌套知识标记平台用到的 Web 技术包括 Web 前端技术和 Web 后端技术也是必不可少的。

第三章　法律知识图谱的本体设计

本章针对构建法律知识图谱所必须的也是最重要的本体设计进行讨论。本章将主要介绍法律知识的文本建模过程。首先，是法律知识的来源；然后，我们将介绍我们设计的法律知识 schema。主要是设计法律知识的实体类型和关系类型有哪些。由于法律知识所具有的嵌套结构，我们又对法律知识进行了关系约束，避免法律知识嵌套结构过于复杂而难以利用。

对于一个可靠的知识图谱来说，其知识源必须是可靠的，也就是说提取知识的来源必须是公众认可的，而不是从互联网等不确定消息源中获取的信息。再者，可靠知识源的知识本体概念必须是是符合知识源的文本内容，并且符合相应领域的基本本体概念的，这要求设计者从知识源文本中进行观察研究，对领域内容进行学习研究，之后才能将文本和领域知识相结合设计出完美的 schema 来。

法律知识结构较为复杂，在科创板上市领域也是一样的。其中富含逻辑概念，是一个与数学较为接近的偏文科的综合学科。从中提炼出逻辑关系，和逻辑变量，是很早就有法学家进行研究，比如道义逻辑。法律道义逻辑中的应当，可以，不能，是模态逻辑的一种表现，不过较贴合现实生活一些。本章将围绕法律道义逻辑及相关法学理论，来对科创板上市业务领域的法律知识进行文本建模抽象出独有的法律知识 schema。

3.1　法律知识源

知识图谱要求知识库中的知识是可靠的。正巧，法律领域有相当可靠的知识源，那就是现行有效的法律法规。这些法律法规都是由专门的立法机构，进行长期的经验积累，形成的适用于我国国情的法律法规。因此，法律法规作为法律知识库的来源是可靠的。但是法律领域，所涉及的法律条文众多，根据中华人民共和国司法部法律法规数据库[①]统计，一共有法规 66637 部。将如此众多的法律条文都转化存储进法律知识库，当然是每个法律人及其研究者的梦想。庞大的法律知识库，由相互交叉的法律规则网络构成，必然有强大的指导，推理能力。但是基于目前的条件和要求，我们只需要对所研究实验的特定领域内的相关法律条文，进行法律知识提取即可。

基于科创板首次公开募股的相关研究，我们的法律数据主要有以下几个来源，

① http://search.chinalaw.gov.cn/search2.html

这里我们只列举其中每一项的一些法规。法律法规：

- 《中华人民共和国证券法》
- 《中华人民共和国公司法》

部门规章：

- 《科创板首次公开发行股票注册管理办法（试行）》
- 《公开发行证券的公司信息披露内容和格式准则第 41 号–科创板公司招股说明书》
- 《公开发行证券的公司信息披露内容和格式准则第 42 号–首次公开发行股票并在科创板上市申请文件》

业务规则：

- 《关于发布 < 上海证券交易所科创板上市公司证券发行上市审核规则 > 的通知》
- 《关于修改 < 上海证券交易所科创板股票上市规则 > 的通知》
- 《关于发布 < 上海证券交易所科创板股票发行上市审核规则 > 的通知》

以上法律数据来源中，我们提取核心的一些规则条文，进行知识提取。因为法律条文中含有一些非规范性的条文，是不具有法律规则属性的，所以就可以省略掉它们。并且法律条文中的一些程序性相关的条文，对解决实体问题作用不大，也可以省略掉一部分，减轻标记任务。

其中我们会重点关注《证券法》中关于公司首次公开发行新股的条件。以及其他法规中关于发行条件，上市条件，是否符合科创板的定位，以及信息披露，中止审核，终止审核等的内容。

3.2 法律知识表示 schema

构建法律知识图谱需要法律知识的本体论做指导，即对法律文本中的知识进行建模。

3.2.1 实体类型

首先定义实体类型的 schema，法律法规是规范人们行为为目的的，故最重要的就是行为实体。人员做出某些行为时，这些行为可以作用到人员、文件以及一些特殊事项等。当人员和行为及其作用物结合在一起时，就构成了某一事项。所有的事项都可以有时间、数值、指标、特征等来约束，修饰它们。最后，否定词可以对事项和行为置否，表示该事项或行为不能满足。如表3–1所示。

表 3–1 法律知识图谱中的实体类别

Table 3–1 Entity category in legal knowledge graph

名称	定义	例子
行为	动作、行动方式	发行人（申请股票首次发行上市）的
人员	可以作为主体的自然人、法人或其他组织等	(发行人)；(保荐人)；
文件	单独的文件、带修饰的文件等	(招股说明书)；(发行人的招股说明书)；(保荐人的上市保荐书)；
代词	指代关系的主语	(下列事项)；
特征	描述性、修饰性的属性	(预先披露）的招股说明书；内容应当(真实)、(准确)、(完整)；
事项	情形、事件	(确需当面咨询的)，可以 · · ·；下列情形之一：(A)、(B)、(C)；下列事项：(A)、(B)、(C)；
时间	时间点、时间段等	本所（收到上市申请文件五个工作日内)；
数值	数字、数字范围、比例等	预计市值（不低于人民币 10 亿元)；
指标	可以衡量的标准	(预计市值）不低于人民币 10 亿元
否定词	不、非等	预先披露的招股说明书等文件（不是）发行人发行股票的正式文件；

3.2.2 关系类型

定义关系类型,需要根据实体类型,及一些常用的逻辑来表示它们。如表3–2所示。

- 表示逻辑运算的两种关系：与、或。与关系表示两个实体之间都必须满足，缺一不可。或关系表示两个实体之间满足其中一个就可以。
- 三种道义逻辑：应当、可以、不能。道义逻辑中应该、允许、禁止等行为逻辑动词。它们对应在法条当中具体表现为：应当、可以、不能等。三种道义逻辑动词的作用域是一样的，所以将它们放在一起。
- 条件为关系：表示行为或事项等发生的前提等。比如："本所收到文件五个工作日内，对文件进行检查"，其中有关系（本所，应当，对文件进行检查)，但是它的条件是 "收到文件五个工作日内"。所以为了结合其条件，该语句

应表示为（(本所，条件为，收到文件五个工作日内)，应当，对文件进行检查)。这种嵌套关系可以比较全面地表达本句所蕴含的意义。

- 是一种关系：表示下位词实体拥有上位词实体的关系。比如“会计师事务所、律师事务所等证券服务机构”中，证券服务机构是一个上位词，会计师事务所和律师事务所都是它的下位词。所以此句可以表达为（会计师事务所，是一种，证券服务机构）和（律师事务所，是一种，证券服务机构)，也可以表达为（(会计师事务所，或，律师事务所)，是一种，证券服务机构)。可以看出这里文本中的“和”实际上表达的是逻辑中的“或”关系。因为会计师事务所和律师事务所这两个实体并不需要同时满足。故所提取出的知识都应是逻辑上表达方式。而不是文本当中字面的意思，需要一定的判断能力。这为关系抽取增加了不少困难。
- 定义为关系：表示某一实体的概念是什么。比如“营业收入：指公司利润表列报的营业收入;”这是一个典型的概念介绍语句。可以表达为（营业收入，定义为，公司利润表列报的营业收入)。法律知识中多有一些法律专有名词，需要有定义为关系来表达。
- 指代关系：表示某一代词实体在文本中指代的具体是哪一个非代词实体。中文文本中常用你，我，他（们）等常用代词来简化语句。法律文本中也一样，还有各种各样的代词。如“下列事项之一：A、B、C”其中下列事项是 A 或 B 或 C 的代词。可以用三元组表示为（下列事项，指代，A 或 B 或 C)，A 或 B 或 C 其实也是一个具有嵌套关系的三元组，具体为（下列事项，指代，((A，或，B)，或，C))。在存储法律知识的过程中，这些代词实体实际上是不具有实际含义的。因此，并不需要存储这些代词，只需将其替换为三元组的宾语即可。
- 同义词关系：表示 A 与 B 是同一个物体，也就是一个物体有两个名字。如“科创板股票上市委员会（以下简称上市委员会)”，可以表达为（科创板股票上市委员会，同义词，上市委员会)。同义词关系，多为“简称”，“又称为”等模式的关系，而不是近义词。近义词多指含义相近而不是相同的词语。近义词与同义词多被混淆，需特别注意。
- 作用于关系：表示一个行为作用在某一物体上。可以是人员、文件甚至是一个行为，一个事项。如“报送下列发行上市申请文件”可以表达为（报送，作用于，发行上市申请文件)。
- 子属性为关系：表示两个实体之间有属性依附关系。比如“人的手”，可以表达为（人，子属性为，手)；“发行人的控股股东”，可以表达为（发行人，

子属性为，控股股东）。不同于“是一种”关系，子属性为不是包含关系，实际上即 A 是 B 的一部分。

表 3–2 法律知识图谱中的关系类别

Table 3–2 Relation category in legal knowledge graph

名称	定义	例子
与	A 与 B 必须同时成立，缺一不可	具备健全且组织良好的组织机构；（健全，与，组织良好）
或	A 与 B 其中一个成立即可	下列情形之一：A、B、C；下列事项：A、B、C；（（A，或，B），或，C）
应当	和义务相对应	”发行人应当按照规定聘请保荐人”：(发行人，应当，按照规定聘请保荐人)
可以	和权利相对应	”发行人可以通过上市审核业务系统进行咨询”：(发行人，可以，通过上市审核业务系统进行咨询)
不能	和禁止性义务相对应	”预先披露的招股说明书等文件不能含有股票发行价格信息“：(预先披露的招股说明书等文件，不能，含有股票发行价格信息)
条件为	条件、前提等	“本所收到文件五个工作日内，对文件进行核查”：((本所，条件为，收到文件五个工作日内)，应当，对文件进行核查)
是一种	下位词和上位词之间的关系	”会计师事务所、律师事务所等证券服务机构“：（会计师事务所和律师事务所，是一种，证券服务机构）
定义为	概念关系	”营业收入：指公司利润表列报的营业收入；“：（营业收入，定义为，公司利润表列报的营业收入）
指代	指代	“下列事项之一：A、B、C”：（下列事项，指代，A 或 B 或 C）
同义词	A 与 B 是同一个东西	”科创板股票上市委员会（以下简称上市委员会）“：（科创板股票上市委员会，同义词，上市委员会）

续下页

续表 3–2

名称	定义	例子
作用于	行为作用于人员、文件等	“报送下列发行上市申请文件”：(报送，作用于，发行上市申请文件)
子属性为	A 是 B 的一个属性或一部分	”发行人的控股股东”：（发行人，子属性为，控股股东）

3.3 法律知识关系约束

实体类别和关系类别的种类一共有 10 种和 12 种，但是加上嵌套结构的表示，它们之间的组合数量就没有上限，实在太多，难免出错。因此要对实体类别和关系类别的连接方式进行约束。

如图3–1所示，右侧一方框中带颜色的箭头表示关系的类型。左侧表示小方块表示实体类型，它们之间交错的箭头表示可以连接到的其他实体。最下面的两个虚线方块表示两个嵌套实体“行为”和“事项”。其中：

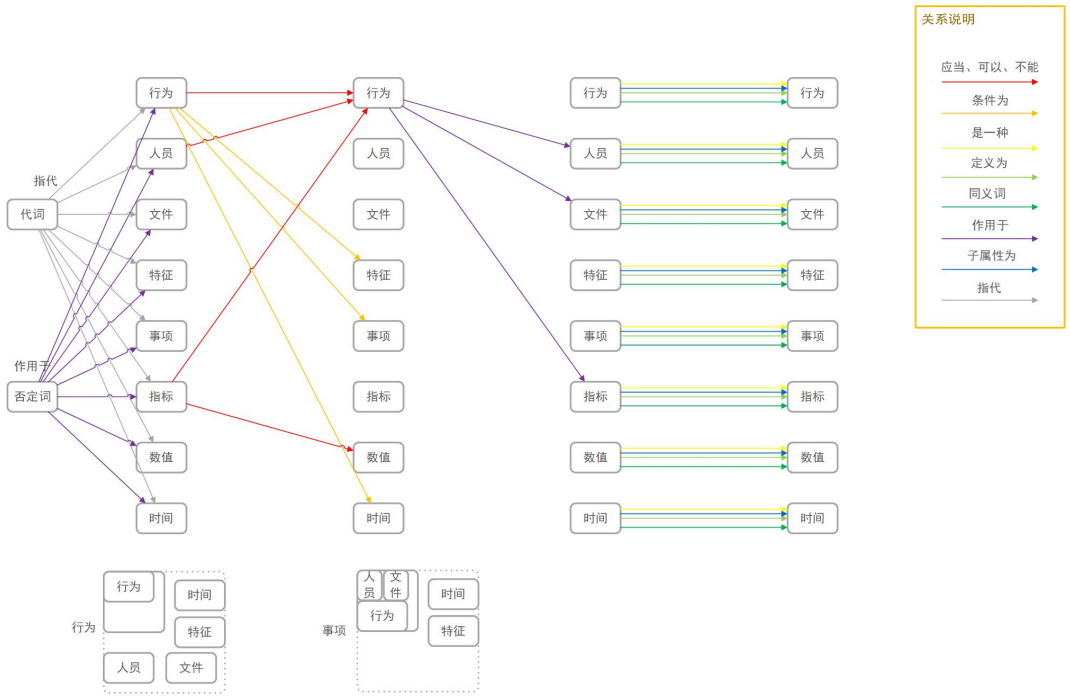

图 3–1 schema 关系约束

Figure 3–1 schema constraint

- “代词”和“否定词”实体类型可以用“指代”和“作用于”关系类型连接到除自身的其他任何实体类型。
- “行为”，“人员”和“指标”可以用道义逻辑动词“应当”，“可以”，“不能”关系类型来连接到“行为”实体类型。“指标”实体类型也可以用道义逻辑动词“应当”，“可以”，“不能”关系类型连接到“数值”实体类型。
- ”行为”实体类型可以用“条件为”关系类型连接到“特征”，“事项”，“时间”等实体类型表示约束。同时“行为”实体类型可以用“作用于”关系类型连接到“人员”，“文件”，“指标”等实体类型。
- “定义为”，“同义词”，“子属性为”等关系类型可以通过除“代词”，“否定词”实体类型的其它实体类型连接到自身。
- “行为”嵌套实体表示一个行为中必须有一个动词类“行为”实体类型，还可以包括“人员”，“时间”，“特征”，“文件”等实体类型。“事项”嵌套实体表示一个事项中必须有一个人员和行为，或一个文件和行为，还可以包括“时间”，“特征”等实体类型约束。

在实际标记过程当中很难区分一个行为或事项到底是该嵌套还是不嵌套。这涉及到标记粒度的问题。在本项目中并不严格区分标记粒度，也就是说一个行为或事项可以标记成一个较短的实体也可以标记成一个较长的大实体，不再体现嵌套实体。标记任务的难度会随着粒度的由大变小逐渐增加。不区分实体长短会对专家标记带来更友好的体验。

3.4 本章小结

本章介绍了法律知识库的 schema 设计。法律知识建模，要求我们寻找可靠的法律知识源，通过分析和科创板上市相关的法律内容，以及法律道义逻辑要求的法律规范，我们设计了一套法律知识表示的 schema。其中包括实体类型和关系类型，为了防止关系嵌套产生大量的冗余知识，我们对法律知识的关系进行了约束，形成一套完善的法律知识图谱 schema。

第四章　基于嵌套知识标记平台的知识抽取

为了提取复杂的法律知识，本章将介绍一个嵌套知识结构标记平台的构建，此标记平台是专门为我们所设计的法律知识结构制作的。实践当中该标记平台应用广泛，为嵌套知识结构的知识抽取提供了高效的工具准备。

知识图谱中存储着人类的知识，但是知识本身是复杂的。很多专家学者都对知识的结构进行过优化整改，这里本文提出了一种嵌套知识结构。

传统的知识结构是二元的，用三元组来表示就是（实体，关系，实体），而嵌套知识结构中实体和关系都有可能是嵌套的。实体是嵌套的意味着实体可能里面包含着另一个实体或多个实体；关系是嵌套的意味着关系的两端中的一端连接的是另外的一个关系，也就是说可能是一个三元组。比如天变黑导致下雨和变凉，就可以用一个三元组表示：（天变黑，导致，（下雨，和，变凉））。为了使嵌套知识结构能更高效快速地提取，本章将详细介绍嵌套知识结构标记平台的构建。

标记平台的构建是为了对法律知识进行提取，本章将利用嵌套知识标记平台进行知识抽取之后，将知识进行了存储，设计了嵌套知识结构的存储模型，形成了一个可靠的法律知识图谱。

4.1　嵌套知识标记平台的需求分析

4.1.1　用户登录与注册

用户首次登录需要注册账号密码，完成信息的录入和收集。同时注册失败，注册成功，登录失败，登录成功等状态显示。注册失败是指由于网络原因，或密码、用户名等不符合要求等。注册成功是指成功地将用户名和密码保存进数据库中。登录失败是指输入信息有误或不存在等信息提示。登录成功是指用户登录进标记平台进入主页面。如图4–1为登录和注册用例图。

4.1.2　任务管理

每个用户有自己的标记任务，有自己的组群。可以查看自己的任务列表，其中包括已经完成的任务和未完成的任务。管理员可以添加、删除任务；可以创建、删除任务群组。如图4–2为任务管理用例图。

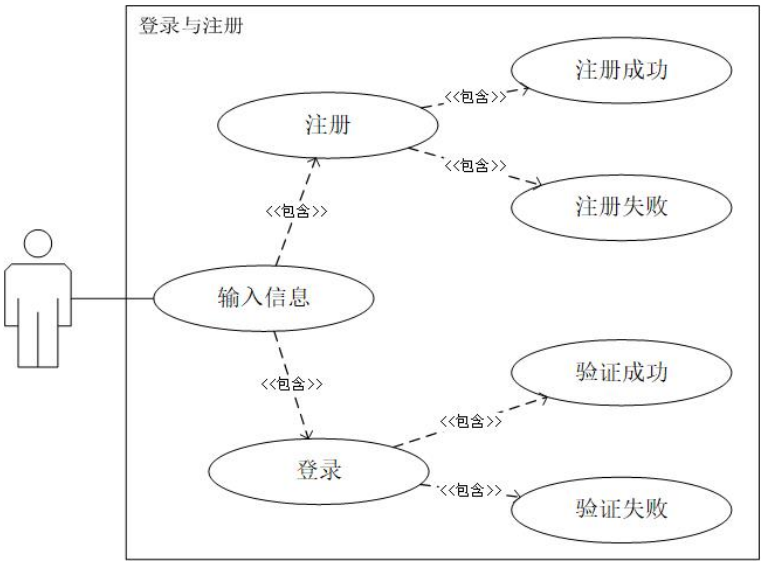

图 4–1 登录和注册用例图

Figure 4–1 Use case diagram of login and register

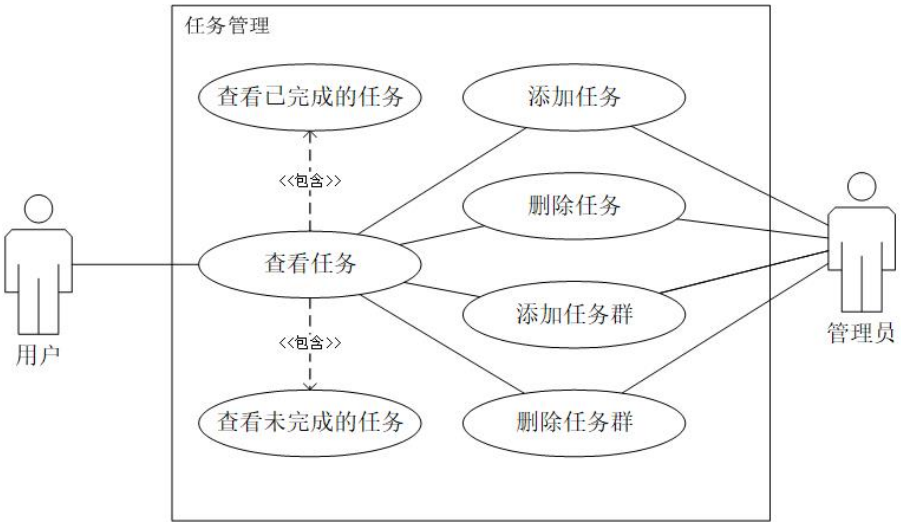

图 4–2 任务管理用例图

Figure 4–2 Use case diagram of task management

4.1.3 标记文档

用户点击一份任务，可以展示标记文档任务。然后用户就可以在展示的标记文档中直观地进行各种标记操作。如图4–3为标记操作的用例图。

具体有如下 3 种主要的标记操作：

- 分词标记

  在展示的标记文档任务中，用户可以分词，合词等操作。合词是分词的逆操作。

- 实体标记

  在展示的标记文档任务中，用户可以选择实体标签进行标记实体，以及对实体的修改，删除等操作。这里的实体是指嵌套实体，具体就是实体里面可以嵌套一个或者多个实体。

- 关系标记

  在展示的标记文档任务中，用户可以选择关系标签进行标记关系，以及对关系的修改，删除等操作。这里的关系是指嵌套关系，具体就是关系两端连接的可以是另一个关系或实体。

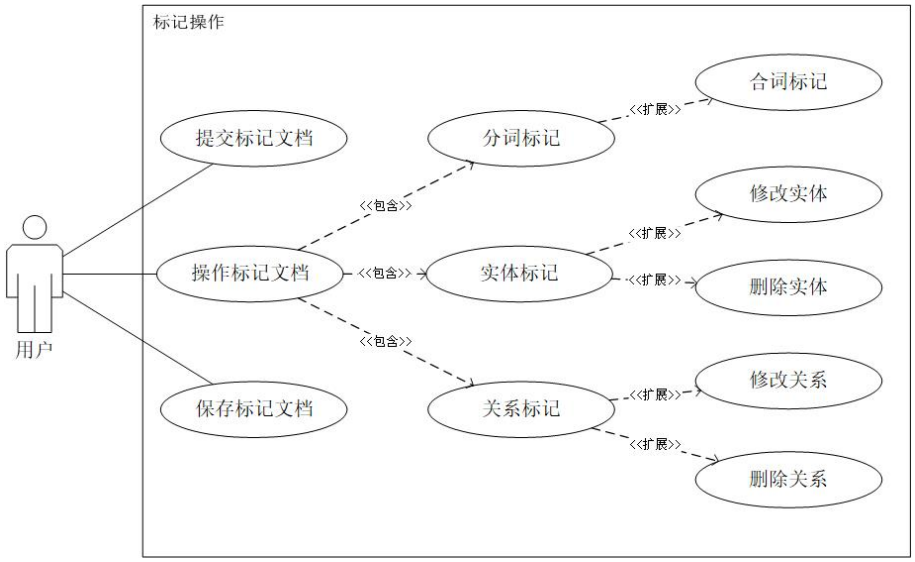

图 4–3 标记用例图

Figure 4–3 Use case diagram of marking

4.1.4 保存提交任务

用户标记完任务文档后可以点击保存按钮提交到数据库中。也可以点击查看按钮查看已经完成的任务文档。点击已完成的任务文档，可以展示已经标记好的标记文档，但是不能修改。如图 4-3，用户操作标记文档的时候可以随时保存和提交标记文档任务。

4.2 标记平台的设计

标记平台的设计主要包括两部分：标记平台的总体架构设计和标记平台的可视化组件设计。

4.2.1 标记平台的总体架构设计

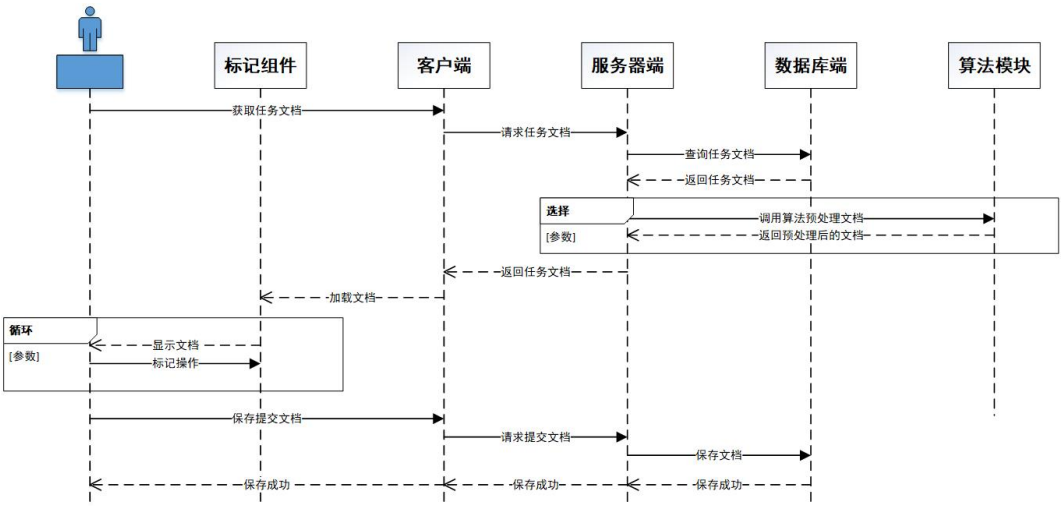

图 4–4 标记平台序列图

Figure 4–4 sequence diagram of marking platform

如图4–4，标记平台包括五大部分：前端用户及任务管理部分，前端标记组件可视化部分，自然语言处理算法部分，后端任务处理部分，数据库管理部分。前端用户及任务管理部分展示用户登录注册页面，通过 http 请求获取用户的个人信息及任务列表。前端标记组件可视化部分在用户选择某一任务时，对该任务进行文档级结构化展示；并通过用户的交互操作，对该文档的标记内容进行更新。自然语言处理算法部分提供 3 个主要功能：分词算法，实体识别算法，关系抽取算法。后端任务处理部分调用自然语言处理算法部分提供的功能将用户选取的任务

文档进行预处理后发送给前端部分进行展示。数据库管理部分就是管理和存放用户名密码和各种任务文档的地方。

4.2.2 标记平台的可视化组件设计

标记平台的可视化组件是标记平台的一个核心组件，是和用户直接交互的部分。如图4–4，标记平台的可视化组件设计是依托于客户端的。由于要具有可以标记嵌套知识结构的实体和关系的能力，标记组件需要考虑的内容有很多个，比如：

- 生成不同种类的标签实体（可嵌套）
- 标签和标签的关系表示
- 标签和关系的关系表示
- 关系和关系的关系表示

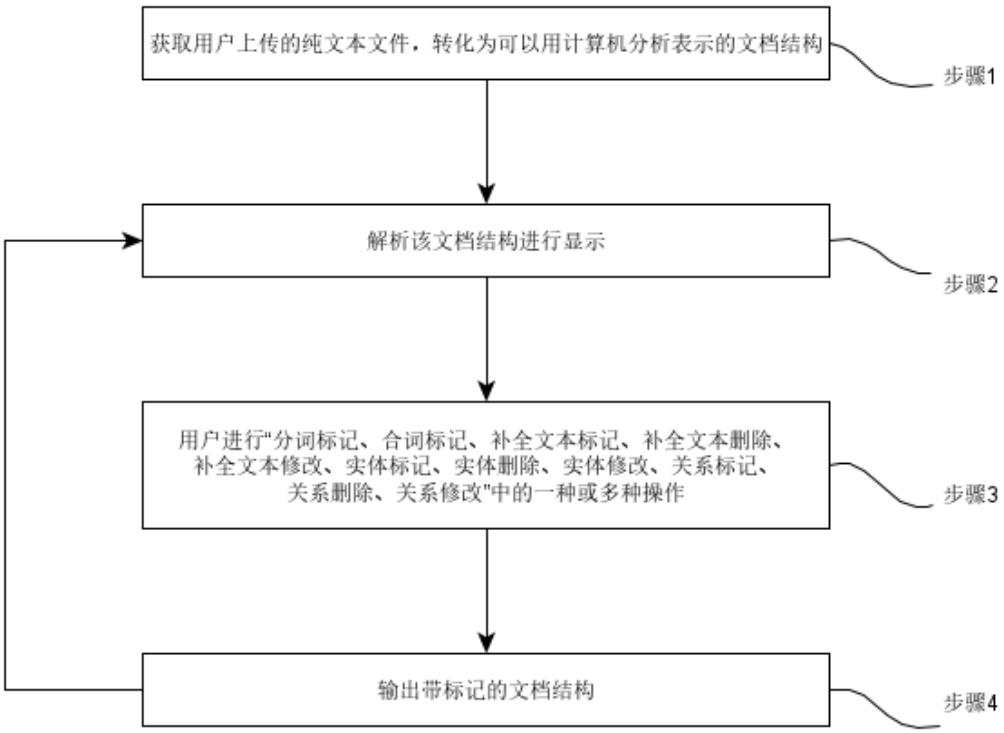

图 4–5 标记方法流程图

Figure 4–5 Flow chart of marking method

实际项目开发过程中，我们不断完善标记平台标记组件的功能，使其功能全备并具有较好的亲用户性。具体标记平台的标记方法流程如图4–5所示：

首先，获取用户上传的纯文本文件，转化为可以用计算机分析表示的文档结

构；其次，解析该文档结构进行显示；再者，用户就可以进行“分词标记、合词标记、补全文本标记、补全文本删除、补全文本修改、实体标记、实体删除、实体修改、关系标记、关系删除、关系修改”中的一种或多种操作；最后，就可以输出带标记的文档结构。当然该文档结构还可以继续被解析，继续让用户标记，直到获取全部的知识。

4.3 标记平台的实现

标记平台的实现在实践当中有众多的参与人。这里我们将详细介绍标记平台的前端和标记组件模块来说明嵌套知识结构的抽取方式，之后简单介绍后端对数据库和自然语言处理算法的集成调用设计。

4.3.1 前端模块

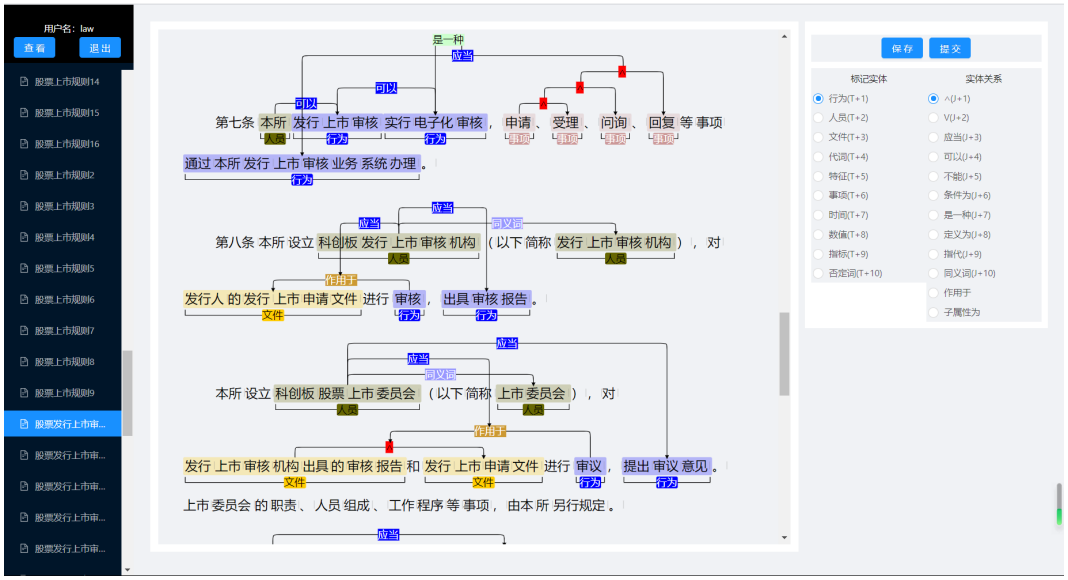

图 4–6 前端设计

Figure 4–6 Front end design

我们共设计了用户登录页，用户注册页，用户信息页，用户任务管理页，用户任务查看页，用户任务标记页等六个主要的界面。如图4–6，为用户任务标记页的前端实现效果图，左边为任务栏和用户信息栏，中间为标记组件模块加载的内容显示，右边为实体和关系标签选择栏。其中任务栏显示待标记的任务，用户信息栏显示用户信息，为了能更方便用户标记和呈现标记效果，标记组件在中间面积最大的部分，实体和关系标签选择栏方便用户直接选择需要标记的标签类别。当

用户完成标记后可以点击保存和提交按钮进行任务提交，用户可以在用户任务查看页查看已经完成的标记任务。

前端采用 React 技术设计 UI 和界面，利用 Restful API 设计交互接口。npm 作为项目的包管理器，webpack 作为打包工具。此技术在第二章 Web 前端技术中有详细介绍。

4.3.2 标记组件模块

标记组件采用 TypeScript 语言，它是一个面向对象的语言，是 JavaScript 的超集。使用第二章 Web 前端技术介绍的 SVG.js 作为标记组件响应交互以及界面元素内容的工具。SVG.js 生成一张画布，我们需要将所有的内容利用 SVG.js 里的 Rect、Path、Text 等元素绘制出来，放到画布上进行操作。

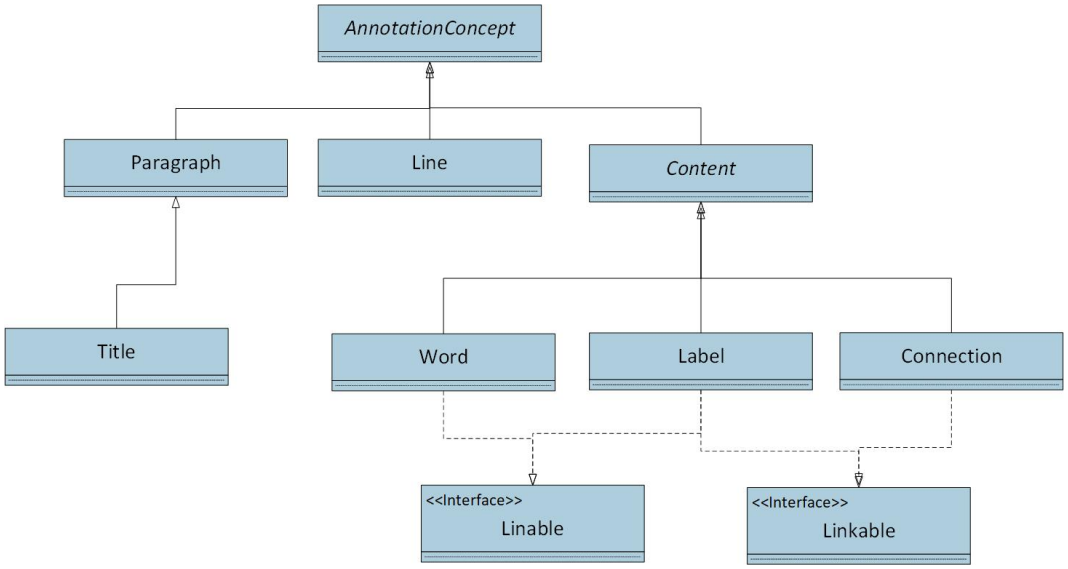

图 4–7 可视化组件的类图

Figure 4–7 Class diagram of visual components

如图4–7，是我们设计可视化组件的类图。所有对象都继承于 AnnotationConcept 这个虚类，Content 也是一个虚类，有 Word 类、Label 类、Connection 类继承自 Content，作为画布的主要内容。Line 类和 Paragraph 类是整合 Content 内容的行和段。Title 类是 Paragraph 类的一个子类。为了支持嵌套，我们设计了两个接口类型：Linable 和 Linkable。Linable 接口表示一个对象可以在一个 Line 里，Linkable 接口表示一个对象是可以被线条 Connection 连接的。

具体来说绘制 Label 和 Word 需要注意的是：

1. Label 和 Word 都是（linable）
2. 绘制它们的同时计算宽度，然后按顺序排列
3. Label 是可递归的，里面的元素是 linable，即可以是 Word 也可以是 Label，绘制的时候需计算好高度

绘制 Connection 需要注意的是：

1. Connection 的两端是 label 或 connection (linkable)
2. Connection 有自己的层级，基于所在的 line（src 或 dst 靠顶端的一个所在的 line）；如果加 connection, 就一一比较所在 line 的 connections 计算出它的层级（区间覆盖，如果该 connection 和比较的 connection 相覆盖就将它的层级 +1）

绘制 Line 和 Paragraph 需要注意的是：

1. Line 是存在于 Paragraph 里的，相当于一个自然段有多行文字。
2. Line 需要依次排布好 Word 和 Label，同时计算好 Line 的高度。
3. Paragraph 需要依次排布好 Line，然后计算好高度。

绘制好了这些元素后，需要有一个 Annotation 类来整体排版，需要注意的是：

1. Paragraph 是存在于 Annotation 里的，相当于一篇文章有多个自然段。
2. Annotation 会依次排布好 Paragraph，然后加完 Connections 。（因为 Connection 会跨段落）
3. 事件函数（加 Label, 消 Label, 加 Connection, 消 Connection, 分词, 合词, 键盘事件），事件的监听由 Word, Label, Connection 等注册；处理由 Annotation 来处理。

最后，我们用游览器的开发者工具的性能里录制火焰图作性能测试。同样，我们采用 npm 作为包管理工具，webpack 作为项目打包工具。

4.3.3 后端模块

后端主要采用 java 语言进行编写，主要是统筹调配和功能管理。后端管理用户信息，和用户的任务，并调用自然语言处理模块提供的分词、实体识别、关系抽取接口进行预标记，最后将用户标记好的文档存储进数据库中。如图4–8所示为后端架构图。

具体来说，标记平台的后端模块有如下一些主要功能：

（1）用户管理模块：用户管理模块包括用户的信息管理，用户的权限管理，与前端的用户管理交互模块相对应。

（2）标记内容管理模块：标记内容管理模块包括标记任务的分发、标记任务

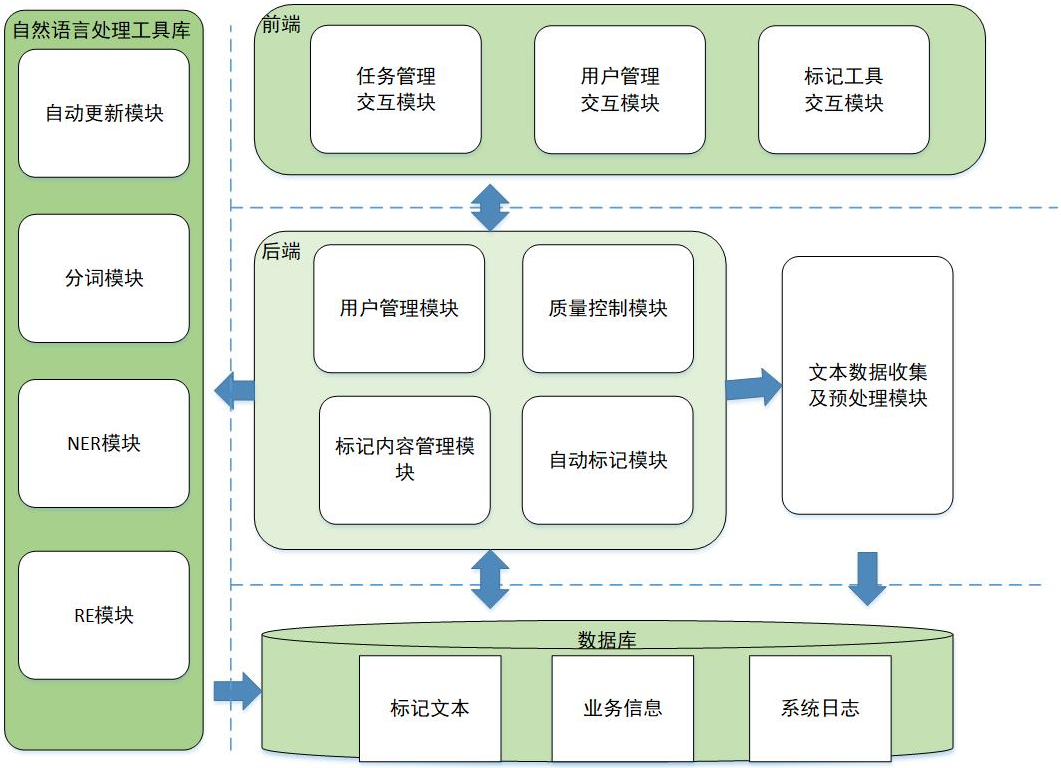

图 4–8 后端架构图

Figure 4–8 Architecture diagram of back-end

组群的创建与删除等。与前端的任务管理模块相对应。

（3）自动标记模块：自动标记模块会调用自然语言处理工具库中的分词模块、NER 模块、RE 模块对未进行标记过得文档进行预标记。这一个模块由项目组其他成员完成，他们不断使用自然语言处理工具库的自动更新模块，调取数据库中的标记完成的文档，来训练模型升级模型。这一模块极大的加快了标记人员的工作效率。

（4）质量控制模块：为了提高标记质量，降低质量控制难度，我们设计了自动的质量控制机制，通过自动化的质量控制方法（包括但不限于重复任务交叉比对、任务规则约束等），可以给出具体标记任务的详细质量评估以及标记人员的水平评价。根据具体标记任务的质量评估，审核人员可以提高审核的效率以及准确度。根据标记人员的水平评价，审核人员可以有针对性地对标记内容进行审核，提高效率并提高正确性。

（5）数据获取模块：数据获取模块设计了数据自动获取机制，通过分析用户需求，从公开和授权的自然语言文本来源（包括但不限于公开的网页、授权数据库

等）获取文本并存入系统供用户选择使用，从而简化了用户获取数据的流程，降低数据获取难度，提升文本数据数量和覆盖面。同时，系统还支持用户上传自定义文本，方便用户自定义标记任务，进一步提升文本覆盖面。

我们使用 maven 做为项目的统一开发规范与工具以及统一管理 jar 包的工具。使用 Restful API 作为前后端交互的接口设计。

服务器端我们使用了 Elasticsearch，MySQL 做我们的数据库，主要存储用户信息，标记文档，业务信息，和系统日志信息。这里不详细介绍存储信息的建模。

自然语言处理工具库由项目组其他人员完成这里不详细介绍模型相关的信息。具体来说，自然语言工具库主要提供 3 个功能接口，即分词接口，命名实体识别（NER）接口和关系抽取（RE）接口，以微服务的方式供我们随时调用。自然语言处理工具库可以调用数据库中的标记文本进行模型训练来升级模型。

4.4 法律知识抽取

法律知识抽取是指将法律知识源中的知识抽取出来。本文搭建了嵌套知识结构的标记平台来专门标记和抽取知识。此处，描述知识抽取的具体方法，整体来说是一个半自动化的方法。首先用标记平台标记少量知识样本，做为训练集，训练 NLP 模型，最终 NLP 模型就可以对大量法律文本进行初始标记了。NLP 技术模块一共提供 3 个模型接口，分别是分词，实体识别 (Named Entity Recognition) 和关系抽取 (Relation Extraction)。分词模型，使用开源的分词模型即可，准确率要求不高，如果加入领域词汇字典，将会使模型准确率得到很大提升。实体识别模型和关系抽取模型由项目组的其他成员完成。其中实体识别模型的效果惊人的好，采取了第二章中的 BiLSTM-CRF 模型。可是关系抽取模型的效果并不理想，也是一个难度系数相当高的任务，因为这里面加入了嵌套知识结构，针对的问题是嵌套关系的抽取。为了保证知识的准确性，现在 NLP 模型并不能直接取代人力，只能帮助人力做好初始标记，最终需要人力来校对审核。不过幸运的是，校对完成的数据又可以作为训练数据提升 NLP 模型的整体效果。本文核心在于可靠法律知识图谱的构建及应用，NLP 相关的技术本文将不再说明。

4.5 法律知识存储

法律知识存储及将法律知识以持久化的方式存储到图数据库 neo4j 中。具体的流程如图1–1所示，需要将标记好的数据转化为 RDF 三元组集，再将 RDF 三元组集导入到图数据库 neo4j 中。

首先，将标记平台标记好的任务转化成 RDF 三元组集。所谓 RDF 三元组即 < 主体 (subject), 谓语 (predicate), 宾语 (object)> 这里我们简单定义了一种字符串结构 (subject,predicate,object), 为了符合嵌套结构的知识表示，这种字符串结构中 subject 和 object 也可以是 RDF 三元组。具体输出的三元组的数据结构为:

- 英文括号表示元组边界
- 英文逗号分隔字段
- 英文冒号分隔实体名称和类型名称
- 主语和宾语字段可以为独立实体，也可以为元组
- 类型为字符串

例如:

> ‘(发行人: 人员, 应当, 业务完整: 事项)‘

> ‘((发行人: 人员, 子属性为, 信息披露: 事项), 应当,((真实: 特征, 与, 准确: 特征), 与, 完整: 特征))‘

其次，将上述转化好的 RDF 三元组集转化为图数据库 neo4j 中对应的实体。对法律知识实体和关系进行实体建模:

```
class NeoEntity:
    def __init__(self,  name, type_, id_=None):
        self.name = name
        self.id = id_
        self.type = type_

class NeoRelation:
    def __init__(self, start_node, rel, end_node, id_=None, source=None):
        self.start_node = start_node
        self.rel = rel
        self.end_node = end_node
        self.id = id_
        self.source = source
```

上述例子对应的建模实体在 neo4j 中存储为下图4–9所示。

实际存储过程中，neo4j 的表示方式只有实体和实体之间连接边，而没有从边连接到边的方式。因此为了能够支持嵌套结构的语义，需要增加一个 “嵌套实体” 的实体类型。

最后，为了提高系统的查询效率，本文在 neo4j 外层包上一个 redis 缓存池。由于 redis 的 key-value 特性，查询效率得到了明显提升。

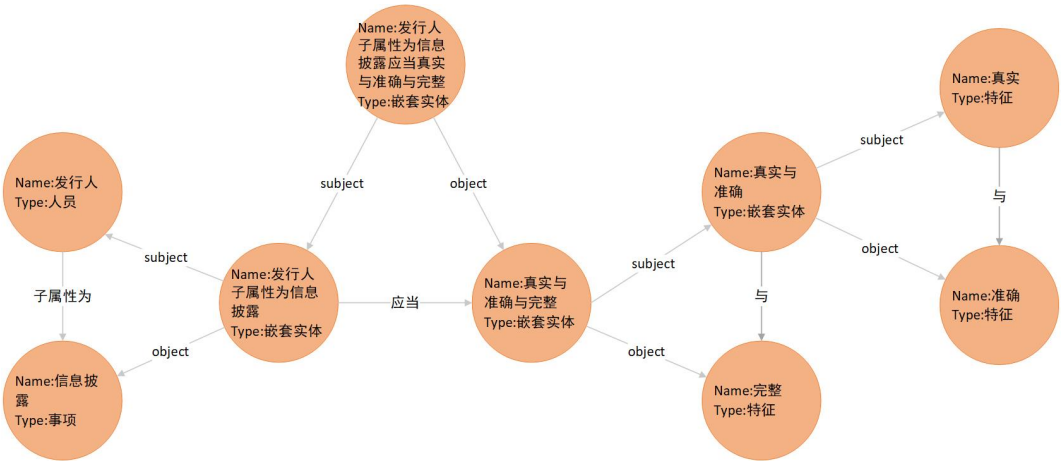

图 4–9 嵌套结构在 neo4j 中的示例

Figure 4–9 Nested structure example in neo4j

4.6 本章小结

本文介绍了嵌套知识结构的标记平台的构建，首先我们提出了嵌套知识结构的表示，然后我们对嵌套知识结构的标记平台进行了需求分析，再者我们设计了嵌套知识结构标记平台的总体架构及其主要的标记组件的可视化设计，最后我们展示了嵌套知识结构标记平台的标记页面，以及可视化标记组件的详细设计思想，并且介绍了后端架构设计以及各个模块的功能。最终，我们利用了标记平台抽取了复杂的法律知识，然后对这些知识进行了存储。

第五章　法律知识图谱的法律规则转化及应用

法律规则，是人类从原始社会进化为部落再到国家，逐步发展形成的社会成果。它并不是人为创智的，而是社会的一面镜子，是社会的反映。如今人们将法律规则收集，并颁布成文法典，使社会更加透明，法治的道路更加宽阔。法治体现了一个国家的文明程度。如何利用好法律规则，如何让法律更好地服务广大人民群众，如何让习近平新时代法治思想深入人心，是一个时代在思考的问题。

本章提出了一个针对可靠的法律知识知识图谱的应用架构，并做了简单的实现，针对的是科创板上市审核结果预测的问题。本章将具体应用标记平台并根据设计好的法律知识建模 schema，将抽取好的法律知识存储到图数据库中后，进行应用转化。具体来说，法律知识图谱的规则转化及应用，即基于法律道义逻辑，法律三段论推理，将具有嵌套知识结构的知识图谱进行对应转化的方法，以及如何收集预测所需要的信息，如何判决推理的应用。

5.1　架构设计

本研究的重点是推理科创板上市的审核结果，所以我们设计一套专门推理能否在科创板上市成功的法律知识应用架构。具体就是，将我们的法律知识图谱进行法律知识的转化和现实生活中存在的事件信息进行结合，最终推理出法律规则的判断结果的过程。其内部设计如图5–1所示，主要有 3 个子模块：

（1）知识转换器（KBConverter）：从法律知识库中抽取出法律规则，并将这些规则转换为道义逻辑公式。

（2）爬虫收集器（Spider）：从互联网中收集爬取公开披露的招股说明书，和其他一些公司披露信息。由于每份招股说明书大概有 500 页 pdf 的 A4 纸。所以将爬取的信息存储在全文搜索引擎 Elasticsearch 中。

（3）判决器（Judger）：根据 Elasticsearch 中的招股说明书的实际信息，提取相关要素要件，作为法律三段论推理的小前提。然后，寻找道义逻辑公式集中的法律规则，作为法律三段论推理的大前提。最后，得出法律规则结果。总体来说，判决器就是一个法官，最终预测能否上市。

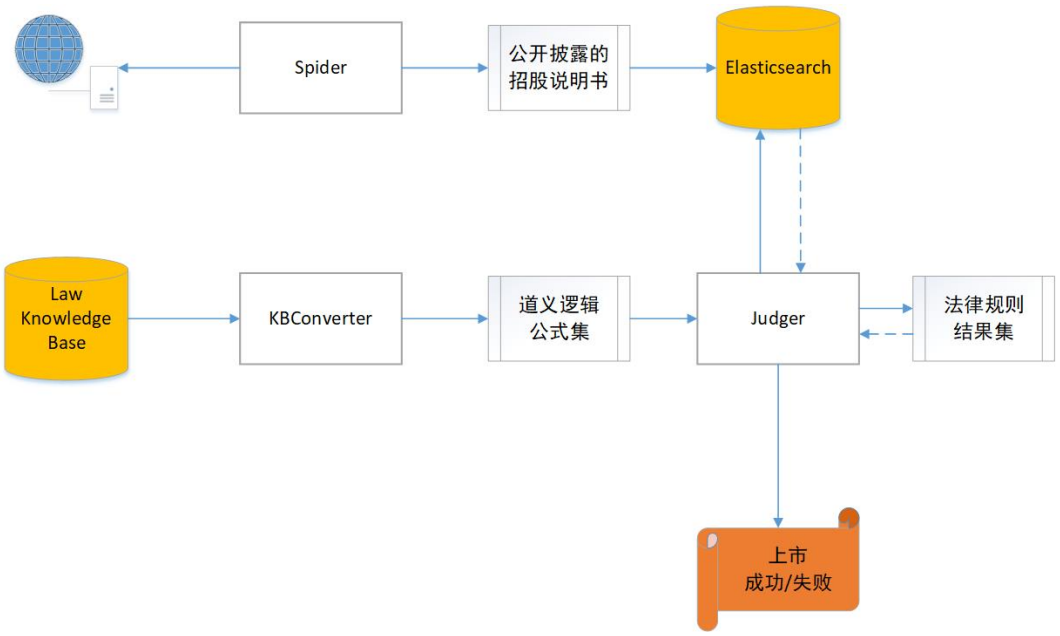

图 5–1 法律知识应用架构图

Figure 5–1 Architecture diagram of legal knowledge application

5.2 法律知识转换器

本节将介绍法律知识转换器的功能及实现方式，其主要功能有两个：1）从法律知识库中获取法律知识；2）将法律知识转化成道义逻辑公式。

5.2.1 法律三段论推理与法律规则的应用

法律知识图谱其目的是构建法律规则的知识图谱，法律规则的运用过程[37]如下：

1. 寻找小前提，即查明案件事实。
2. 寻找大前提，即找到与案件事实相对应的法律规范。
3. 从大前提和小前提中推理出符合立法者的目的的法律结果。

具体来说，法律规则是国家强制力机关规定人们的法律权利，法律义务和相应的法律后果的一种行为规范。其中，法律规则有一定的结构形式，它要求任何法律规则都由三个部分组成：假设条件，行为模式和法律后果。

（1）假定条件

假定条件是基于人民的行为作出假设，如果发生了某种行为，就怎么样的一种推理模式的前提。

（2）行为模式：

行为模式包括三种：可为模式，应为模式，勿为模式。分别指在某一假定条件发生时，人们“可以”，“应当或必须”，“禁止或不得”怎么行为的行为模式。

（3）法律后果：

法律后果包括：合法后果和违法后果。

基于以上分析我们采用了将法律规则转化为传统的 if-then-else 的模式。或者说是基于经典道义逻辑的规则表达方式：

可为模式：

$$if\ p\ then\ q \Leftrightarrow pP \rightarrow q \tag{5–1}$$

应为模式：

$$if\ p\ then\ q \Leftrightarrow pO \rightarrow q \tag{5–2}$$

勿为模式：

$$if\ p\ then\ q \Leftrightarrow pF \rightarrow q \tag{5–3}$$

其中：O 表示“应该”，P 表示“允许”，F 表示“禁止”。p 表示 CNF 合取范式（conjunctive normal form），q 表示 DNF 析取范式（disjunctive normal form）。

5.2.2 提取相关法律知识

我们构建的法律知识库可以看成是一张 RDF 语义网络图 G。根据道义逻辑动词的主语实体出发，提取相关法律知识，就是从 G 中提取一张子图。提取相关知识子图需要一个参数：道义逻辑的主语实体集 $S_{morality}$。而根据道义逻辑动词的主语实体集获取法律知识子图为所有法律知识子图的并集，即

$$\begin{aligned} G_{marolity} &= \bigcup_{s \in S_{morality}} G_s \\ &= \bigcup_{s \in S_{morality}} DFS(G, s, \Phi) \end{aligned} \tag{5–4}$$

为了获取法律知识，这里我们使用深度优先搜索图 G 来提取道义逻辑动词的主语实体的相关知识子图 G_s，程序的主要目的是以道义逻辑动词的主语实体 s 为出发节点在图 G 上使用深度优先搜索算法，不断扩张图 G_s，直到不能扩张为止。深度优先搜索算法 DFS 如附录中算法A–1所示，其中包含 s 的最大嵌套实体

Clause 表示包含 s 同时仅有“与”、“或”关系的一个庞大的实体。如图5–2所示，实体节点 t_2 是实体节点 a, b, c, t_1, t_2 的最大嵌套实体 Clause, 实体节点 d 的最大嵌套实体 Clause 就是其本身。

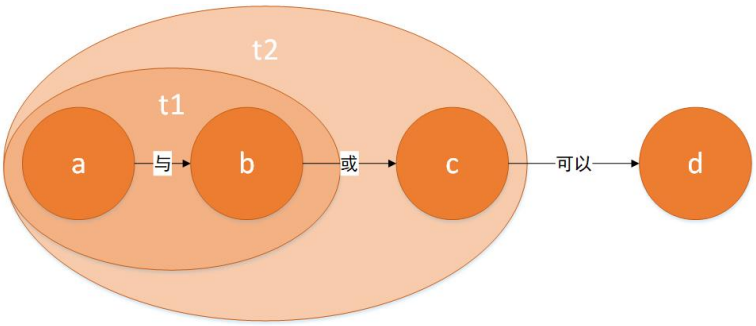

图 5–2 顶层子句嵌套实体示例

Figure 5–2 Example of top clause recursive entity

5.2.3 道义逻辑公式编码

拿到实体的知识子图后，就可以将实体的知识子图编码成道义逻辑公式集合。具体来说，就是将实体的知识子图中的 RDF 三元组转化成道义逻辑公式。具体过程有如下两部分：

（1）把 RDF 三元组表示的主体和客体分别编码成道义逻辑变量。这是一个双射的对应表，比较简单，比如“真实，准确，完整”分别映射成“d_1, d_2, d_3”。

（2）把 RDF 三元组中间的关系编码成道义逻辑符号。

这里，我们会将映射好的逻辑变量，编码成道义逻辑规则 R。为了推理方便，我们将规则的形式设定为 $(v_{l_1} \wedge v_{l_2} \wedge ... \wedge v_{l_m})M \rightarrow (v_{r_1} \vee v_{r_2} \vee ... \vee v_{r_n})$。比如将“$(d_1 \vee d_2)M \rightarrow s$”拆解为两条道义逻辑规则“$d_1 M \rightarrow s, d_2 M \rightarrow s$”。其中，$M$ 为道义逻辑动词符号：“O, P, F”。

对应我们设计的 schema，我们使用普通的逻辑变量编码实体，道义逻辑联结词编码关系。其中实体类型的编码是双射对应表，没有特别的约束。如表3–2为道义逻辑联结词编码关系的对应表。

表 5–1 法律知识图谱中的关系类别

Table 5–1 Relation category in law knowledge graph

关系名称	道义逻辑联结词	例子
与	与 ($\wedge$)	(健全，与，组织良好）编码为“健全 $\wedge$ 组织良好”
或	或 ($\vee$)	((A，或，B)，或，C）编码为“$A \vee B \vee C$”
应当	应为 ($O \rightarrow$)	(发行人，应当，按照规定聘请保荐人) 编码为“发行人 $O \rightarrow$ 按照规定聘请保荐人”
可以	可为 ($P \rightarrow$)	(发行人，可以，通过上市审核业务系统进行咨询) 编码为“发行人 $P \rightarrow$ 通过上市审核业务系统进行咨询”
不能	勿为 ($F \rightarrow$)	(预先披露的招股说明书等文件，不能，含有股票发行价格信息) 编码为“预先披露的招股说明书等文件 $F \rightarrow$ 含有股票发行价格信息”
条件为	与 ($\wedge$)	((本所，条件为，收到文件五个工作日内)，应当，对文件进行核查) 编码为“(本所 $\wedge$ 收到文件五个工作日内）$O \rightarrow$ 对文件进行核查”
是一种	蕴含 ($\rightarrow$)	(会计师事务所和律师事务所，是一种，证券服务机构）编码为“会计师事务所和律师事务所 $\rightarrow$ 证券服务机构”
定义为	蕴含 ($\rightarrow$)	(营业收入，定义为，公司利润表列报的营业收入）编码为“营业收入 $\rightarrow$ 公司利润表列报的营业收入”

续下页

续表 5–1

关系名称	道义逻辑联结词	例子
指代	在知识库存储时已经将“a 指代 b”中的 a 用 b 替换，所以此关系再知识库中不存在	————
同义词	等价 (↔)	(科创板股票上市委员会，同义词，上市委员会）编码为“科创板股票上市委员会 ↔ 上市委员会”
作用于	会将“a 作用于 b”关系合并成一个实体 ab	(报送，作用于，发行上市申请文件) 编码为“报送发行上市申请文件”
子属性为	会将“a 子属性为 b”关系合并成一个实体“a 的 b”	(发行人，子属性为，控股股东）编码为“发行人的控股股东”

5.3 爬虫收集器

爬虫收集器是收集信息，获取法律推理中小前提的必然工具之一。如何从众多的网络信息中找到有用的案件事实，是互联网法律科技的一个重点问题。

公司 IPO 上市需要大量人力物力，一个 IPO 项目需要持续 1-3 年，科创板注册制极大加快了上市的进度，最快的仅需几个月就可以完成。但是总体来说，公司需符合各种法律法规成为一个合格的股份有限责任公司，还是需要一年半载的时间。上市项目一旦启动就要将公司各方面的信息披露，这里就方便我们做信息收集的工作了。结合公司 IPO 上市的相关业务，我们主要的信息需求有：招股说明书，及其他公司上市项目的相关信息。

为此，我们设计了一套简单的爬虫系统，如图5–3所示:

爬虫收集器将要收集的信息的链接放入调度器，等待爬取，当爬虫收集器没有工作时就取来一个链接，发包给网络，下载器收报，解析器解析，最后通过管道放入到内容管理系统，然后爬虫收集器继续拿取链接，直到所有的链接都被爬虫收集器收集完毕为止。我们选取 ELK（Elasticsearch,Logstash,Kibana）套装作为内容管理系统。内容管理系统是方便人工审核的后台管理系统。

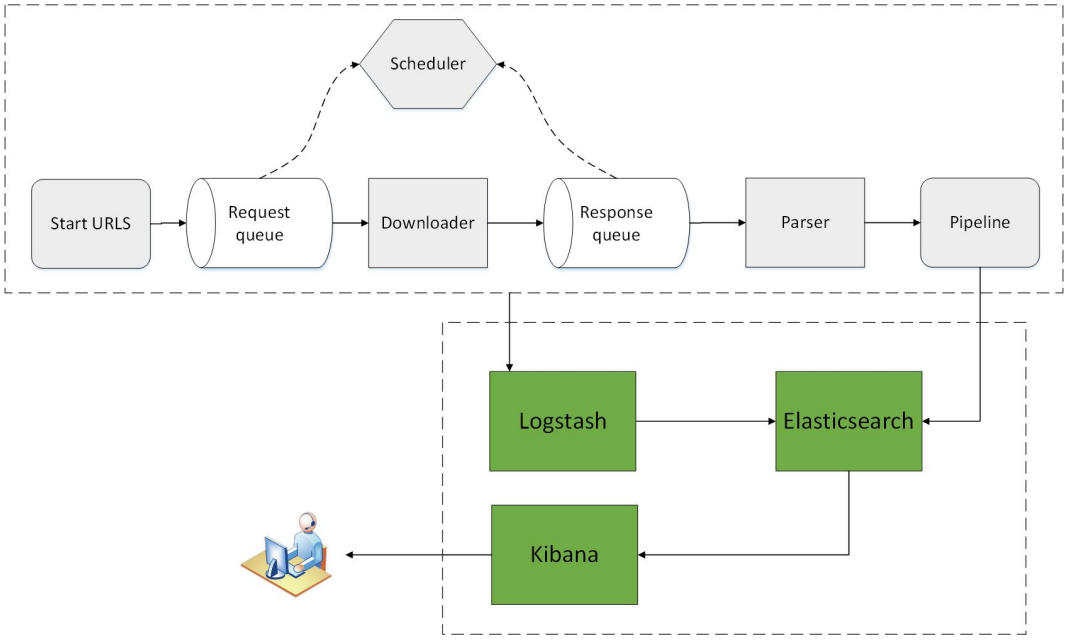

图 5–3 爬虫架构图

Figure 5–3 Crawler architecture diagram

5.4 判决器

判决器（Judger）充当一个法官的角色。它将运用法律推理三段论对科创板能否上市进行“自由心证”，得出最终结果。如图 5-1 所示，判决器将集合招股说明书和其他公司上市项目的相关信息作为认定案件的小前提，将道义逻辑公式集作为认定案件事实的大前提，最后再综合全部的法律规则结果集得出上市成功或失败的结论。

5.4.1 大前提的获取

针对大前提的获取我们根据已经转化好的道义逻辑公式集，逐一进行匹配结合。具体就是搜集所有的道义逻辑实体集合。然后将道义逻辑实体集合中的每一个实体，进行法律知识转换器的抽取与编码。实际当中，我们按照本文嵌套知识结构的提取方法进行法律知识提取。目前，我们的法律知识库中有 428 个实体和 900 个关系。但是，被法律专家进行剔除后只剩下 260 个实体和 229 个关系。如表5–2和表5–3所示，实体和关系类型的分布如下：

表 5–2 实体类型分布

Table 5–2 Entity type distribution

实体类型	占比	数量
行为	4.62%	12
人员	25.38%	66
文件	1.54%	4
代词	0.77%	2
特征	10.00%	26
事项	51.15%	133
时间	2.69%	7
数值	1.54%	4
指标	2.31%	6
否定词	0.00%	0

表 5–3 关系类型分布

Table 5–3 Relation type distribution

关系类型	占比	数量
与	27.07%	62
或	31.00%	71
应当	10.04%	23
可以	1.75%	4
不能	6.11%	14
条件为	12.23%	28
是一种	4.37%	10
定义为	0.00%	0
指代	0.00%	0
同义词	0.00%	0
作用于	2.62%	6
子属性为	4.80%	11

据此我们进行了对应的法律道义逻辑公式编码，共获得法律道义逻辑公式 Rules: 102 和变量 Variables: 130。这些法律道义逻辑公式就是逻辑形式上的法律规则，并被法律专家认为是正确可靠的。接下来，我们就可以进行法律规则的判断，并依据这些法律规则作为法律推理的大前提。

5.4.2 小前提的获取

针对小前提的获取我们根据道义逻辑，应为，可为，勿为三种模式设计了三种查询接口：分别对应 Elasticsearch 中 query 查询[①]的”must”,”should”,”must_not” 字段。并且幸运的是 Elasticsearch 中提供的”analyzer” 字段可以进行语义分析。这样我们就可以轻松的从中提取招股说明书及其他上市相关信息的事实，作为规则判断的小前提了。为此我们设计判断规则是否成立的算法，在附录中算法B–1：给出一份招股说明书和一条规则，判断规则是否成立，或者条件是否满足。

5.4.3 预测结果

判决器准备好了大前提和小前提，并对每一条规则进行判断获得了规则结果集。那么它可以基于这些根据进行预测。据此，我们提出了 2 种预测内核：

（1）理想预测内核：该内核认为法律是严格的，即每一条规则都是要被满足的。但是基于目前信息获取方面能力的不足，以及语义理解，价值判断等不确定性，该内核不具有实践可能性。

（2）分类器内核：该内核认为法律是经验构成的，即有大多数合法即合法。目前，我们也是采用该内核进行预测的，即分类器 svm。实验证明我们并不需要多么完美的分类器就可以根据这些规则结果集推理出一个较为不错的结果。下一章将是我们的实验部分。

5.5 本章小结

本章介绍了法律知识图谱的规则转化及应用的详细内容。首先，我们介绍了法律知识应用的架构。接下来，我们分模块逐一介绍了法律知识转换器，爬虫收集器，和判决器。法律知识转换器，将法律知识进行抽取并将它们转换为法律道义逻辑公式。爬虫收集器，将预测科创板上市项目的各种信息和内容进行了提取收集。判决器，将法律规则和提取的法律事实进行判断得出法律规则结果集，然后进行预测结果。

① https://www.elastic.co/guide/index.html

第六章　对科创板首发上市审核结果的预测

科创板（The Science and Technology Innovation Board; 简称为 STAR Market）实行注册制，是国家主席习近平 2018 年末宣布设立的一个上海证券交易所的板块。这里我们简单介绍一下注册制的相关规则。

根据《上海证券交易所科创板首次公开发行股票发行与上市业务指南》，上市公司需要转型为股份有限公司，在科创板申请注册，经上海证券交易所的上市委会议同意后，再经证监会同意注册，就可以在注册后一段时间内自己选择上市发行股票的时间点。之后将是初步询价和网上网下申购等具体的流程了。因此，一家上市公司的审核结果的成功与否关键在于证监会是否同意注册。

同时，该发行与上市业务指南还给出了如图6–1的股票发行申请、发行准备及发行流程图：

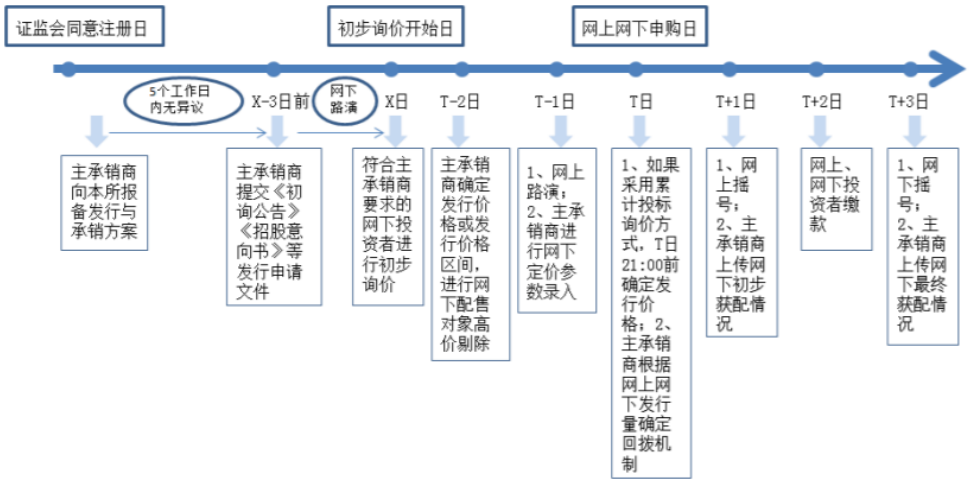

图 6–1　股票发行流程图

Figure 6–1　Stock issuance flow chart

从以上发行与上市业务指南和科创板网站①公布的其他规则信息分析可知，公司在经证监会同意注册后，就可以按照指定流程进行发行上市了。至于发不发行，此时选择权在于公司了。因此一个公司上市成功的关键点在于是否能够经证监会同意。而在经证监会同意之前也需要历经申报稿、上会稿、注册稿等阶段不

① http://star.sse.com.cn/

断被问询与回复。如图6–2为一个最快在科创板通过注册的公司。即使这样也需要数个步骤，逐一通过才行。

图 6–2 上市申请注册流程图

Figure 6–2 Listing application registration flow chart

这里，我们分别收集了上市公司项目的申报稿、上会稿、注册稿的招股说明书，作为我们预测的基本数据来源。

6.1 科创板首次公开募股数据集

上一章中如图 5-1，判决器（Judger）得到了一份法律规则结果集。也就是说，每一份招股说明书都会生成一份法律规则结果集。我们分别收集了 2020 年 6 月和 2020 年 12 月的科创板中的招股说明书，其中包括申报稿、上会稿、注册稿。并且获得了当时的公司上市审核状态信息，如表6–1所示。其中注册结果阶段中，注册生效的公司占全部的 99% 以上，因此可以使用注册结果的码值作为证监会同意注册上市的表示。

表 6–1 公司上市状态码

Table 6–1 company listing status code

状态	码值
已受理	1
已问询	2
上市委会议结果	3
提交注册	4
注册结果	5
中止	7
终止	8

接下来我们将 2020 年 6 月收集的 678 份招股说明书，和 2020 年 12 月收集的 1038 份招股说明书分别用判决器（Judger）进行判决得到对应的 678 份法律规则结果集和 1038 份法律规则结果集。如下表6–2所示，我们将这些法律规则结果集

按八二开分为训练集和测试集。

表 6–2 数据集信息

Table 6–2 data set information

时间	训练集	测试集
2020 年 6 月	542	136
2020 年 12 月	830	208

6.2 预测算法

6.2.1 支持向量机算法

本文采用人工智能领域的支持向量机算法（Support Vector Machine, SVM）作为预测算法。具体来说科创板上市是否成功就是预测公司能否顺利进入到注册结果阶段。基于此分析，对科创板上市审核结果的预测问题就转化为公司能否顺利进入到注册结果阶段的问题。也就是公司的审核状态码是否为 5 的二分类问题。更确切的说就是，给定法律规则结果集 $X = \{X_1, ..., X_n\}$ 和审核状态码 $y = \{y_1, ..., y_n\}$，这里 $n = 102$ 对应 102 条法律规则，其中每个样本都包含多个特征，从而构成特征空间（feature space）：$X_i = [0, 1, -1]$ 分别代表法律规则的条件不成立，法律规则成立和法律规则不成立，而学习目标为二元变量 $y \in \{-1, 1\}$ 表示负类（negative class）即审核状态码不为 5，和正类（positive class）即审核状态码为 5。

本文的预测模型为：

$$f(\theta) = min_\theta C \sum_{i=1}^{m} [y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)})] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2 \tag{6–1}$$

其中 $cost$ 为核函数，$\frac{1}{2} \sum_{i=1}^{n} \theta_j^2$ 为正则项，C 为正则化参数。

6.2.2 核函数

在某种情况下一些线性不可分的问题却在高维希尔伯特空间（Hilbert space）H 里很可能是非线性可分的，这时，分开样本的就是一个超曲面（hypersurface），通过映射到该高维空间中，可以将问题转化为较为简单的线性可分的问题。超平面可以用公式表示如下：

$$w^T \phi(X) + b = 0 \tag{6–2}$$

式中 $\phi : \chi \rightarrow H$ 为映射函数。其中映射函数的形式复杂，且计算其内积及其复杂，此时可使用核方法，也就是说，将映射函数的内积定义为核函数（kernel function）：$K(X_1, X_2) = \phi(X_1)^T \phi(X_2)$ 来巧妙地回避内积的计算问题。

线性核函数（Linear Kernel）是支持向量机算法的核函数之一，擅长于线性分类，它是最基本的核函数。公式如下：

$$K(X_1, X_2) = X_1^T X_2 \tag{6–3}$$

多项式核函数（Polynomial Kernel）也是支持向量机算法的核函数之一，类似于多项式公式，公式如下：

$$K(X_1, X_2) = (X_1^T X_2)^n \tag{6–4}$$

高斯核函数（Gaussian Kernel），是支持向量机算法的核函数之一，使用者众多。在 SVM 中它还有一个名字叫做径向基核函数（Radial Basis Function, 简称 RBF）。公式如下

$$K(X_1, X_2) = exp(-\frac{\| X_1 - X_2 \|^2}{2\sigma^2}) \tag{6–5}$$

Sigmoid 核函数（Sigmoid Kernel）也是支持向量机算法常用的核函数之一，多用作激活函数，公式如下：

$$K(X_1, X_2) = \tanh [a(X_1^T X_2) - b] \quad a, b > 0 \tag{6–6}$$

本文用到的核函数（kernel）有：‘linear’，‘poly’，‘rbf’，‘sigmoid’。

6.2.3 正则化参数

本文采用 L_2 正则化的传统方法来防止模型过拟合（Overfitting）。L2 参数正则化是最常用的正则化方法之一，也常常是机器学习算法中默认选择的正则化方法。

本文用到的正则化参数 C 有：0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000。正则化的强度与 C 成反比。必须严格为正。

6.3 评估标准

本文采用分类任务中较为常用的精确率（Precision，P），召回率（Recall，R），F1 值（F1 score，F1）对实验结果进行评估。这三个指标的计算公式分别如下三个

公式所示：

$$P = \frac{TP}{TP + FP} \times 100\% \tag{6–7}$$

$$R = \frac{TP}{TP + FN} \times 100\% \tag{6–8}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{6–9}$$

对于公司的招股说明书来说，TP 代表了正确预测了该公司能否上市的数量；FP 代表了预测成功而实际没有成功的数量；FN 代表预测没有上市成功但是实际上已经上市成功的数量。本文在预测上市成功与否的最终评价标准为 F1 值。

6.4 结果分析

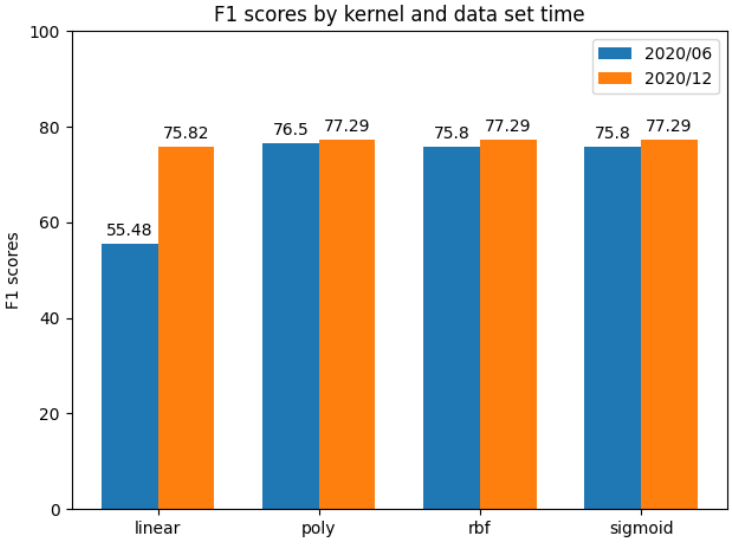

图 6–3 实验结果

Figure 6–3 Experimental results

2020 年 6 月份的数据和 2020 年 12 月份的数据本质上没有什么区别，但是数据量上相差近一倍。如图6–3即使数据量都是几百个数据，可以看出我们的模型预测的结果 F1 值还是不错的。2020 年 6 月份的数据集和 2020 年 12 月份的数据集整体 F1 值几乎没有相差。说明数据量并不是模型的关键点。

最终我们得到了最好的结果是:

- 2020 年 6 月数据：核函数 kernel=poly, 正则化参数 C=1，F1=76.50%.
- 2020 年 12 月数据：核函数 kernel=poly, 正则化参数 C=0.01，F1=77.29%.

虽然我们的数据量很少，但是由于我们的数据都是知识推理过后的法律规则结果集。所以，小数据加上简单分类器也能得到了还不错的结果。

6.5 本章小结

本章我们简单介绍了科创板注册制的规则，并且根据我们从判决器（Judger）中得到的法律规则结果集设计了制定了模型预测的策略。然后，我们介绍了模型的各种信息，包括核函数、正则化参数等。接下来，我们采用了分类任务中常用的评估函数 F1 值。最后，我们的实验结果显示，预测结果 F1 值还不错的，说明我们的法律规则结果集具有一定的可靠性。

第七章　总结与展望

7.1　全文总结

本文主要研究了法律人工智能在科创板上市领域的新应用。本文介绍了此研究的研究背景及意义，然后，介绍了法律相关的国内外现状，发现目前法律领域较为传统，且现行多是端到端的模型架构，不足以支撑法律所要求的可解释性，本文提出了一套法律知识抽取，法律规则转换与应用的架构，并在科创板上市审核模块做了实验研究。具体来说，本文的主要贡献有：

首先，我们设计了一套可推理的法律知识库的知识表示 schema。为了法律知识的文本建模，我们寻找了可靠的法律知识源，即法律法规的条文。其次我们根据法律知识的结构设计了一套可以推理法律知识的 schema，其中有实体类型十种，有关系类型十二种，这里面蕴含了法律道义逻辑推理关键内涵。最后我们对复杂关系进行了关系约束。

其次，我们构建了一个嵌套知识结构的标记平台并利用其进行了知识提取，将提取的知识进行了存储。基于我们提出的嵌套知识结构的表示，我们对嵌套知识结构的标记平台进行了需求分析，设计了嵌套知识结构标记平台的总体架构及其主要的标记组件的可视化设计。我们展示了嵌套知识结构标记平台的标记页面，以及可视化标记组件的详细设计思想，和后端开发的相关内容。之后我们对知识进行了半自动化的抽取，以及用自己定义的 RDF 三元组做了存储设计。

再者，我们针对法律知识进行了法律规则转化及相应的规则判断。本文设计了一套基于科创板上市的法律知识应用架构。其中知识转换器，详细描述了法律三段论推理与法律规则的应用，并将法律知识进行抽取，转换为了法律道义逻辑公式。然后，我们又利用了爬虫收集器发现推理运用的小前提事实。最后，我们将大前提法律规则和小前提事实喂给判决器做预测推理。判决器果然生成了一份法律规则结果集。

最后，我们对科创板首发上市审核结果进行了预测，这也是本文的目的所在。本文简单介绍了科创板上市的相关规则。基于判决器生成的法律规则结果集，我们采用了人工智能的 SVM 算法作为我们的推理内核。实验结果表明，科创板首发上市审核结果预测 F1 值近 80%，说明我们的提取的法律知识是有用的，我们的法律知识应用框架是有效的。

7.2 未来展望

我们提取了法律知识，构建了法律知识库，搭建了法律知识应用框架，对科创板上市审核结果进行了简单的预测。但是由于时间的限制，本文在多个方面具有不足之处。在未来的工作中，我们会更加贯彻习近平新时代法治思想的内容，对以下几个方面进行研究：

首先，法律知识库的规模扩充。目前，法律知识库中的实体和关系数量很少，只有几百个，虽然已经有效地预测了科创板上市审核的结果，但是知识的精确度，和有效度还有待验证。之后，我们会对法律知识库的内容进行扩充，对法律知识的精度进行提炼。

其次，本文采用的是分类器推理内核，相较于理想法律推理内核还有很大距离，这也正是我们需要努力的目标和方向。我们需要不断完善法律推理机制，使法律建设数字化，信息化，可计算化。

再者，本文在推理预测结束之后，由于时间关系，没有给出预测结果的可解释性说明。其实，可解释性可以在法律规则结果集中很容易找到，这是一个法律系统架构需要显示的地方。之后，我们会增加可解释性结果的说明信息，增加用户信服度。

最后，针对不同法律领域，可以尝试使用本文中的法律知识框架进行应用研究。本文的法律知识应用框架具有通用性，之后我们会在民法、刑法、行政法等领域进行试验研究。让法律科技更加智能。

附录 A　算法 5-1

算法 A–1 获取单一道义主语实体相关知识子图 DFS 函数伪代码

Input: 知识库中的知识图 G，某个道义主语实体 s，已经访问过得节点集合 $Visited$

Output: 道义主语实体 s 相关的知识子图 G_s

```
1  Function DFS(G, s, Visited):
2      G_s ← ϕ;
3      if s in Visited then
4          return G_s;
5      else
6          for e in G_E And (e = (s, s′) Or e = (s′, s)) do
7              if s′ in Visited then
8                  Continue;
9              else
10                 add s′ to Visited;
11                 if type(e) == 应当 Or 可以 Or 不能 Or 是一种 Or 同义词 Or 定义为 then
12                     将 s′ 加入 G_s 的点集，将 e 加入 G_s 的边集;
13                     G_s ← G_s ∪ DFS(G, s′, Visited);
14                 else if type(e) == 条件为 Or 子属性为 Or 作用于 then
15                     T ← e 对应的上层嵌套实体节点
16                     add T to Visited;
17                     将 s′, T 加入 G_s 的点集，将 e 加入 G_s 的边集
18                     G_s ← G_s ∪ DFS(G, T, Visited);
19                 else if type(e) == 与 Or 或 then
20                     T ← e 的顶层 Clause 嵌套实体节点
21                     将 s′, T 加入 G_s 的点集，将 T 中的每一条边都加入 G_s 的边集
22                     add T to Visited;
23                     G_s ← G_s ∪ DFS(G, T, Visited);
24                     for t in T do
25                         add t to Visited;
26                         G_s ← G_s ∪ DFS(G, t, Visited);
27                     end
28                 end
29             end
30         end
31         return G_s;
32     end
```

附录 B　算法 5-2

算法 B–1 规则判断伪代码

Input: 招股说明书的名字 *doc_name*，某一条规则 ***R***

Output: 规则成立 1，或规则不成立-1，或规则的条件不成立 0

```
must_list ←Φ ;
should_list ←Φ ;
must_not_list ←Φ ;
rule_left, morality, rule_right = R ;
must_list.extend(rule_left) ;
if search(must_list) == true then
    if morality == O then
        must_list.extend(rule_right)
    else if morality == P then
        should_list.extend(rule_right)
    else if morality == F then
        must_not_list.extend(rule_right)
    end
    if search(must_list, should_list, must_not_list) == true then
        return 1
    else
        return-1
    end
else
    return 0 ;
end
```

参考文献

[1] Zhuang Y t, Wu F, Chen C, et al. Challenges and opportunities: from big data to knowledge in AI 2.0[J]. Frontiers of Information Technology & Electronic Engineering, 2017(1): 3-14.

[2] Rabelo J, Kim M Y, Goebel R, et al. A Summary of the COLIEE 2019 Competition[C]. in: JSAI International Symposium on Artificial Intelligence. 2019: 34-49.

[3] Branting K, Weiss B, Brown B, et al. Semi-supervised methods for explainable legal prediction[C]. in: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. 2019: 22-31.

[4] Tolan S, Miron M, Gómez E, et al. Why machine learning may lead to unfairness: Evidence from risk assessment for juvenile justice in catalonia[C]. in: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. 2019: 83-92.

[5] Van Doesburg R, Engers T. The False, the Former, and the Parish Priest An Hohfeldian Perspective on Marriage in the Code of Canon Law (long version)[C]. in: 2019.

[6] Prakken H, Sartor G. Law and logic: A review from an argumentation perspective[Z]. 2015. DOI: 10.1016/j.artint.2015.06.005.

[7] Domingos P, Lowd D. Unifying logical and statistical AI with Markov logic[J]. Communications of the ACM, 2019, 62(7): 74-83. DOI: 10.1145/3241978.

[8] Li J, Zhang G, Yan H, et al. A Markov Logic Networks Based Method to Predict Judicial Decisions of Divorce Cases[C]. in: 2018 IEEE International Conference on Smart Cloud (SmartCloud). 2018: 129-132.

[9] El Ghosh M. Automation of legal reasoning and decision based on ontologies[D]. 2018.

[10] Zhong H, Guo Z, Tu C, et al. Legal Judgment Prediction via Topological Learning[C]. in: Proceedings of EMNLP. 2018.

[11] Luo B, Feng Y, Xu J, et al. Learning to predict charges for criminal cases with legal basis[J]. ArXiv preprint arXiv:1707.09168, 2017.

[12] Zhong H, Xiao C, Tu C, et al. How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence[Z]. 2020. arXiv: 2004.12158 [cs.CL].

[13] Duan X, Wang B, Wang Z, et al. Cjrc: A reliable human-annotated benchmark dataset for chinese judicial reading comprehension[C]. in: China National Conference on Chinese Computational Linguistics. 2019: 439-451.

[14] Wang Z, Wang B, Duan X, et al. IFlyLegal: A Chinese Legal System for Consultation, Law Searching, and Document Analysis[C]. in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations. 2019: 97-102.

[15] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. ArXiv preprint arXiv:1301.3781, 2013.

[16] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. ArXiv preprint arXiv:1810.04805, 2018.

[17] Chalkidis I, Fergadiotis M, Malakasiotis P, et al. LEGAL-BERT: The Muppets straight out of law school[J]. ArXiv preprint arXiv:2010.02559, 2020.

[18] Chalkidis I, Androutsopoulos I, Aletras N. Neural legal judgment prediction in english[J]. ArXiv preprint arXiv:1906.02059, 2019.

[19] Verheij B. Formalizing arguments, rules and cases[C]. in: Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law. 2017: 199-208.

[20] 政 李. 科创板发行上市审核制度变革的法律逻辑[J]. 财经法学, 2019: 84-94.

[21] 静 冷. 科创板注册制下交易所发行上市审核权能的变革[J]. 财经法学, 2019: 95-112.

[22] Schneider. E W. Course Modularization Applied: The Interface System and Its Implications For Sequence Control and Data Analysis.[J]., 1973.

[23] Eder J S. Knowledge graph based search system[Z]. US Patent App. 13/404,109. 2012.

[24] Hogan A, Blomqvist E, Cochez M, et al. Knowledge Graphs[Z]. 2020. arXiv: 200 3.02320 [cs.AI].

[25] McGuinness D L, Van Harmelen F, et al. OWL web ontology language overview[J]. W3C recommendation, 2004, 10(10): 2004.

[26] Horrocks I, Patel-Schneider P F, Van Harmelen F. From SHIQ and RDF to OWL: The making of a web ontology language[J]. Journal of web semantics, 2003, 1(1): 7-26.

[27] Føllesdal D, Hilpinen R. Deontic logic: An introduction[G]. in: Deontic logic: Introductory and systematic readings. Springer, 1970: 1-35.

[28] 田雪. 道义逻辑与道义悖论研究[D]. 武汉: 华中科技大学, 2013.

[29] Sun M, Chen X, Zhang K, et al. Thulac: An efficient lexical analyzer for chinese[Z]. 2016.

[30] Sun J. Jieba chinese word segmentation tool[Z]. 2012.

[31] Luo R, Xu J, Zhang Y, et al. PKUSEG: A Toolkit for multi-domain Chinese word segmentation[J]. ArXiv preprint arXiv:1906.11455, 2019.

[32] Manning C D, Surdeanu M, Bauer J, et al. The Stanford CoreNLP natural language processing toolkit[C]. in: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. 2014: 55-60.

[33] Li J, Sun A, Han J, et al. A survey on deep learning for named entity recognition[J]. IEEE Transactions on Knowledge and Data Engineering, 2020.

[34] 杨品莉, 谢志长. 基于 BiLSTM-CRF 的司法领域实体识别研究[J]. 现代计算机, 2020(25): 3-8.

[35] Geng Z, Chen G, Han Y, et al. Semantic relation extraction using sequential and tree-structured LSTM with attention[J]. Information Sciences, 2020, 509: 183-192.

[36] Fielding R T, Taylor R N. Architectural styles and the design of network-based software architectures[M]. University of California, Irvine Irvine, 2000.

[37] 张文显. 法理学[M]. 北京: 高等教育出版社; 北京大学出版社, 2007.

致　谢

感谢交大给我上学的机会，感谢学校教给我做人的道理，感谢导师李国强老师和 Daniel 老师无微不至的照顾，感谢大师兄王若愚给我很多帮助，感谢实验室的小伙伴丁如江，李总，Minkow，罗艺康，钟绿波，谭锦豪，杭媛，以及师弟沙群皓，谭淳给我很多帮助，感谢我的父母给我学费，生活费，感谢生活，感谢天，感谢地。

攻读学位期间获得的科研成果

[1] 第一发明人，“一种在自然语言文本上进行标记的方法”，专利申请号 202010595674.6

上海交通大学

学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：马振文

日期：2021年 1月 6日

上海交通大学

学位论文使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。

本学位论文属于 ☑公开论文

☐内部论文，☐1 年/☐2 年/☐3 年 解密后适用本授权书。

☐秘密论文，____年（不超过 10 年）解密后适用本授权书。

☐机密论文，____年（不超过 20 年）解密后适用本授权书。

（请在以上方框内打“√”）

学位论文作者签名：马振文　　指导教师签名：李

日期：2021年 1月 6日　　日期：2021年 1月 6日

:8001962

118037930075

上海交通大学 硕士 学位论文答辩决议书

姓 名	马振文	学 号	118037930075	所在学科	软件工程
指导教师	李国强	答辩日期	2021-1-6	答辩地点	电信群楼3-318
论文题目	基于法律知识推理对科创板上市审核结果的预测				

投票表决结果：3/3/3 （同意票数/实到委员数/应到委员数） 答辩结论：☑通过 ☐未通过

评语和决议：

马振文同学的硕士学位论文《基于法律知识推理对科创板上市审核结果的预测》，通过构建符合法律推理逻辑的可靠的法律知识图谱，抽取知识，收集数据，对科创板首发上市审核结果进行预测，说明了法律知识应用框架的有效性。较为完整地完成了一个系统的工作。研究方法适当，工作量饱满。

答辩委员会认为：该生选题具有法律科技应用意义；论文报告清晰流畅，回答问题快速、准确；反应出作者具有系统性的基础理论知识和扎实的专业知识，具有独立从事科学研究的能力。

经无记名投票，同意通过了马振文同学的硕士学位论文答辩，建议授予马振文同学专业硕士学位。

2021年 1月 6日

答辩委员会成员	职务	姓名	职称	单位	签名
	主席	邓玉欣	教授	华东师范大学	
	委员	董笑菊	副教授	上海交通大学电子信息与电气工程学院(计算机系)	
	委员	朱其立	教授	上海交通大学电子信息与电气工程学院(计算机系)	
	秘书	庄颖		上海交通大学	