天津大学

本科生毕业论文

题目：面向知识图谱构建的区块链溯源方法研究

学　　院　智能与计算学部

专　　业　计算机科学与技术

年　　级　2018

姓　　名　梁展溥

学　　号　3018216294

指导教师　许光全

独创性声明

本人声明：所呈交的毕业设计（论文），是本人在指导教师指导下，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，本毕业设计（论文）中不包含任何他人已经发表或撰写过的研究成果。对本毕业设计（论文）所涉及的研究工作做出贡献的其他个人和集体，均已在论文中作了明确的说明。本毕业设计（论文）原创性声明的法律责任由本人承担。

论文作者签名：

年　　月　　日

本人声明：本毕业设计（论文）是本人指导学生完成的研究成果，已经审阅过论文的全部内容。

论文指导教师签名：

年　　月　　日

摘　要

互联网时代，信息数据的急速膨胀和数据的异构化发展，使得搜索引擎需要借助语义网络等方式改进检索质量。知识图谱因其结构化、具备高效的推理能力而被广泛运用于智能搜索，推荐系统等领域。但是知识图谱的完整性依赖于广泛和庞大的数据，自动化爬取和开放性提供信息采集接口，导致了知识图谱在构建中存在信息溯源、确权等多方面问题。

因此，本文提出了一种利用区块链技术实现知识图谱的构建和溯源的方法，在提高知识图谱的安全性的同时，降低对系统的性能影响。首先，本文详细分析了现有的知识图谱结构和架构，根据主流的知识图谱应用提出了一种通用的知识图谱构建模型，该模型通过信息抽取、知识融合、加工、推理步骤实现知识图谱的构建。然后基于这个模型，为了提高耦合度，优化了区块链的结构和共识机制，为知识图谱构建过程中出现的信息转化提供轨迹跟踪功能。

在实验环节，验证了本文提出的通用知识图谱构建模型和基于区块链的信息溯源的正确性，实现了知识图谱的构建和溯源。同时，还通过与不同共识机制的区块链系统对比，得出本文提出的方法在资源消耗和性能效率上都有一定优越性。

关键词：知识图谱，区块链，可信，数据溯源，网络安全

ABSTRACT

In the Internet era, with the rapid expansion of information data and the development of data heterogeneity, making it necessary for search engines to improve retrieval quality by means of semantic networks and other means. Knowledge graphs are widely used in intelligent search, recommendation systems and other fields because they are structured and have efficient reasoning capabilities. However, the integrity of the knowledge graph relies on extensive and huge data, and automated crawling and openness in providing information collection interfaces lead to various problems in the construction of the knowledge graph, such as information traceability and confirmation of rights.

Therefore, this paper proposes a method of using blockchain to realize the construction and traceability of knowledge graphs, which improves the security of knowledge graphs while reducing the performance impact on the system. First, this paper analyzes the existing knowledge graph structure and architecture in detail, and proposes a general knowledge graph construction model based on mainstream knowledge graph applications, which realizes the construction of knowledge graph through information extraction, knowledge fusion, processing, and inference steps. Then based on this model, the structure and consensus mechanism of the blockchain are optimized in order to improve the coupling degree and provide a trajectory tracking function for the information transformation that occurs during the knowledge graph construction.

In the experiment, the correctness of the generic knowledge graph construction model and blockchain-based information tracing proposed in this paper are verified, and the construction and tracing of the knowledge graph are realized. Also, by comparing with blockchain systems with different consensus mechanisms, it is concluded that the method proposed in this paper is superior in terms of resource consumption and performance efficiency.

KEY WORDS: Knowledge Graph, Blockchain, Trustworthiness, Data Traceback, Network Security

目 录

第一章　绪论

随着互联网的飞速发展，互联网中的信息急速膨胀，为了在庞大而复杂的互联网数据中根据需要检索信息，搜索引擎应运而生。而随着信息的结构越来越复杂，传统搜索引擎难以满足用户需求。在这样的背景下，谷歌公司提出了知识图谱，用以优化搜索引擎返回结果，增强结果关联性和搜索质量[1]。然而知识图谱在经历多年发展后仍然存在溯源、隐私以及可信等问题[2]，这些问题可能会带来数据安全、信息泄露等风险。

1.1　研究背景及意义

知识图谱在 2012 年被谷歌公司正式提出，最开始是为谷歌搜索引擎提供关联信息的支持，提高搜索结果的准确度和范围。如今，知识图谱的应用已经非常广泛，在智慧搜索、个性化推荐、自动化问答等方面都起到了非常重要的作用[3][4]。

知识图谱是一个语义网络[5]，可以揭示不同实体之间的关系，这使得在检索数据和发掘相似或相关对象时变得更加简单和快速。比如在搜索引擎的场景中，传统的关键词匹配方法得到的结果过于单一，相关度也不高。而且如今用户的搜索行为也逐渐变得越来越复杂和多样化，不仅如此，用户输入的检索请求也会存在模糊的可能性，这就需要搜索引擎需要根据检索语句，对用户所需的目标和场景进行推测，在返回的结果中，不仅要提取精准的信息结果，还要在特定场景的范围内适当发散关系，返回相似的检索结果，提高用户搜索的命中率[6]。例如，用户可能会检索某个大学的官网，那么返回结果除了学校的网址外，还可能就会存在这个大学内的一些学院的官网，以及名称相近的大学的官网（因为可能存在两个学校关系十分密切，比如合作办学等）。知识图谱的产生，能使得现实中的实体和关系映射到计算机世界中，让这些经过处理的数据更加易于使用、计算和维护。

但是，一个足以支持搜索引擎优化的知识图谱系统是十分庞大的，覆盖范围需要足够广，而且时间跨度也十分长。在这里就带来了许多问题，比如已有的数据是如何得到的，古老的历史数据是否依然有效且正确，新加入的信息如何保证正确性等。知识图谱需要长时间且持续地更新才能保证其反映的真实世界的正确性，而历经多次的系统更迭中一定存在标准更改和数据冲突的问题，比如在如今越来越重视的隐私问题中，如何在将信息去敏后再加入知识图谱中，以及如何找

出已有信息的隐私信息部分。随着技术的发展，大数据采集和推荐算法使得人们能在跨越多个不同的场景得到内容相似的推送，这固然在一定程度上方便了用户获取信息，但也带来了在互联网上“裸奔”的担忧[2]。除此之外，采集的信息不一定是准确无误的，如何追溯数据的来源，或根据来源确定数据在知识图谱中的位置，从而剔除或修改知识图谱中的知识，这些都是亟待解决的问题。

而近年来，区块链作为一种新型技术，发展迅猛并逐渐走入人们的视野中。区块链技术于 2008 年伴随着比特币被提出[7]，比特币是一种加密的虚拟货币，因其安全、匿名的特性使得广受欢迎。随着区块链技术的发展，区块链的思想与比特币分离开来，有了更加多的实现，虚拟货币已经只是区块链技术的一种实例。伴随着相关开发工具的完善和标准的制定，区块链技术因其安全、防篡改和可追溯的特性，已经应用于工业制造、商品溯源等领域[8]。比如在奢侈品行业中，类似红酒和收藏品可以附加唯一标识码，通过对商品的信息上链，可以确定商品的来源和经历的所有地方，用以防伪验证和质检时追溯问题来源的依据。

区块链技术使用了加密哈希算法、非对称加密算法保证了基于数学上的安全性[9]，同时又使用了链状结构的存储方式且在算法思想上保证只在末尾增加区块，使得数据在上链后就不可篡改且随时可以追溯。除此之外还使用了去中心化的思想，通过多个节点互相监督，防止少数节点被攻破而导致数据丢失或篡改。这些保障使得区块链技术成为一种解决信任问题的探索，在人与人的往来中可以不再担心信任问题，实现安全可信的交易。

在如今，区块链技术的可靠性已经通过众多虚拟货币的检验，对于该技术在安全和可信领域的发展也有了一定的基础，那么深入探索解决知识图谱中的可信构建和溯源问题的研究也具有重要意义。

1.2 国内外研究现状

1.2.1 知识图谱

早在 1968 年 M. Ross Quillian 就提出了一种基于事物概念和状态以及事物之间的关系的网络[10]，并称之为语义网络，这时的语义网络主要用于计算机理解自然语言的用途。而后又历经万维网的发展，和语义互联网的提出，同时互联网数据的膨胀和人们需求的提高，促使谷歌公司在 2012 年提出了知识图谱的概念，并基于知识图谱优化其搜索引擎，提升检索质量。通过知识图谱优化的语义检索，可以获取更加精确的结构化信息，相比以前的关键词匹配检索，极大提高了搜索结果的准确性和完备性，也提高了用户的使用体验。知识图谱在关系表达和语义提取上有得天独厚的优势，所以不仅在智能搜索领域，还在自动问答系统、推荐

系统、异常监控和风险控制等领域有非常广泛的应用[6]。

因此，在谷歌提出知识图谱的概念后，这就成为了一个热门研究内容。目前国内的像百度知识图谱，不仅自己基于自动化爬取的数据建立知识图谱，供百度搜索使用，同时还开放平台，提供接口给其他企业接入，自行上传数据提升用户的搜索体验和利用百度庞大的数据量来优化自己的产品。而像搜狗百科则向所有人开放编辑，在原本基于页面词条的形式记录信息的基础上，还为各个词条提取实体，在词条间建立知识图谱，提高了词条信息的完整度，更加方便用户检索词条信息。在国外，谷歌不仅最先提出并建立知识图谱，还利用其彻底革新搜索方式，明确提出“搜索不是搜索字符串，而是搜索事物”[11]，将传统的检索关键词转化为从语境中提取实体然后检索实体间的关系，从而得出与实体关联度最高的搜索结果，而这也更加符合用户的搜索目的。除此之外，不同于谷歌的通用型知识图谱，微软根据不同类别单独建立知识图谱，比如 Microsoft Academic Knowledge Graph (MAKG)[12]，这是一个专注于学术性知识的知识图谱，主要针对论文、期刊、作者、机构和研究内容等信息，目前已经收录了超过 2 亿篇科学论文，超过 80 亿的数据。这样的专业型知识图谱收录的信息会更加详细深入和更具专业性，在使用的时候不会被无关信息干扰，准确性更高。

知识图谱是一个知识数据库，保存着实体之间的结构关系，在数据层上来看，保存知识内容是通过三元组数据来实现的，即$< p_1, r, p_2 >$，$p1$和$p2$为实体，r为关系[6]，这样的功能和结构决定了知识图谱在知识处理和应用中的侧重点和效率。特别是在需要联想、模糊检索的领域，应用知识图谱会使得计算机具有类似自然语言推理的能力，而且能形式化地表达事物的关系，直观地发现隐藏的联系，比如在案件侦查和金融领域，犯罪人员通常会通过把联系抹除或多个跳板来隐藏主体与犯罪事件的关系，而知识图谱则可以构建各个实体之间的关系，方便办案人员进行侦查和演绎推理[13]。

1.2.2 区块链

区块链概念最开始是来源于 2008 年中本聪创造的比特币[7]。比特币是一个虚拟货币[14]，即相比于物理货币（如人民币、美元等）来说是没有实体的，由计算机数据构成，且不由任何国家的银行发行。比特币设计之初，是希望解决交易中的匿名和信任问题，所以比特币的设计是去中心化且加密的。每个账户数据同时在多个节点中存在，如果恶意节点需要修改数据，那么就必须要攻破更多的节点才能使数据有效，从而增加攻击的代价，阻止恶意节点对整个系统的破坏[15]。而匿名化则是基于数学理论上的加密算法保证的，比如大数分解和哈希算法，这些数学思想和算法虽然都是基于当前算力下的“难以”解决，但是历经十几年的

发展和验证，同时也是由于比特币的高度稳定性和可靠性，使得这些加密算法的可靠性得到一定证明[9]。

而随着比特币的发展，区块链技术开始从比特币中抽象提取出来，人们不仅在原有比特币的基础上实现了更加多的虚拟货币，比如 Ethereum、Monero，还深入探究了区块链在金融领域之外的行业的应用，比如产地溯源、日志记录、价值证明等领域[16]。得益于其安全性，即不可篡改的特性，区块链技术在许多涉及信任问题的领域都可以进行应用。在国内，百度超级链开放平台[17]提供了可信存证、电子合同、数字身份认证等功能，在版权、金融、溯源等领域融合区块链，提供可信认证和知识证明，比如在百度百科的历史版本页面即可看到每一个版本的更新交易和所属区块信息，保证历史修改记录不会被恶意篡改，而且提供可信的数据溯源能力。在国外，像 IBM 公司则给企业提供基于区块链的可信解决方案，而不是直接运营一个区块链平台[17]，如汽车公司 Renault 和一些农业企业采取区块链的方式解决上下游不同企业之间原材料和商品的可信溯源的问题。可以说，得益于区块链的安全性，区块链技术已经开始广泛应用于企业之间解决信任问题，并正逐渐通过制定标准和更新工具，使其能更方便地解决整个社会交往之间的信任问题。

1.3 论文工作

本文的主要工作是首先研究知识图谱的原理和结构，然后深入探究知识图谱的构建过程，确定不同类型的知识图谱的构建方法和底层结构，总结不同知识图谱构建和维护过程中涉及的数据流通和操作类型，建立一个比较通用的知识图谱模型。然后再探究区块链技术的内部原理，剖析现有的 XuperChain、IBM Blockchain 等实例，根据以太坊的区块链模型，改编共识机制和智能合约，构建一套适用于知识图谱的可信构建和溯源的区块链系统。最后，再根据对接知识图谱系统和区块链系统，将知识图谱的构建和维护生命周期内的操作上链处理，并提供可视化的溯源查询，追踪数据的来源和信息。

1.4 论文结构

本文介绍了面向知识图谱构建中的信息的区块链溯源的方案。

在第一章中，简单介绍了知识图谱的概念，以及其在不同领域中的应用方案，同时提出存在的问题。然后介绍了区块链的概念和特点，研究区块链在解决知识图谱的问题的可能性。

在第二章中，详细介绍了知识图谱的起源和发展，不同行业内解决方案实例

和特点，然后介绍区块链技术的来源和底层依赖，解释安全性的依据，并展示了现有的一些企业应用。

在第三章中，通过对知识图谱和区块链技术的总结，设计了一个通用型的知识图谱构建模型，和一个服务于本方法的区块链系统，用以解决知识图谱的可信构建和溯源问题。

在第四章中，主要描述了实验的设计和实验过程，分析实验的结果，验证本文提出的方法的正确性和性能效率。

第五章是总结和展望。

第二章　相关理论及技术

为了解决知识图谱的可信构建和溯源问题，我们首先应该构建一个通用的知识图谱模型，这个模型需要尽可能地具备一般性，具有大多数知识图谱的构建和维护过程，这样才能使本文的工作更加具备兼容性，方便移植到具体的知识图谱平台。

2.1　知识图谱

2.1.1　知识图谱概述

知识图谱是由众多$< p_1, r, p_2 >$三元组构成的，其中p为知识图谱中的实体，r为知识图谱中的关系，整个知识图谱由实体集合$P = \{p_1, p_2, \dots, p_n\}$和关系集合$R = \{r_1, r_2, \dots, r_m\}$构成的，为 n 个不同的实体，其中这些实体之间有 m 个不同的关系。在这里，实体的定义为对现实世界中的事物的表示，比如具体的某个电影等，关系的定义为两个实体之间的联系的表示，比如参演、导演等。三元组集合$S = P \times R \times P$包含了所有的知识图谱中的三元组，而三元组也是使用知识图谱的重点，正是这样的三元组才提供了推理和查询功能。值得注意的是，这里三元组中的关系 r 是单向的，逆向的关系需要一个新的三元组来表达。除了实体和关系之外，知识图谱中还有两个很重要的内容，即属性和概念。属性是指实体或关系所具有的属性、特点、参数，比如姓名：张三、国籍：中国等。概念是一系列具有相同类型的实体的类别，例如张三和李四都是人。因此，知识图谱内总的来说具有两种数据结构：实体和关系，这同时也是知识图谱的底层数据结构。

根据应用的范围，知识图谱也分成通用型知识图谱和专业型知识图谱[18][19]，通用型知识图谱在广度上更加大，尽可能涵盖更多的实体，比如搜索引擎需要的知识图谱就需要这种类型的；而专业型知识图谱更加注重深度，需要更加详细的属性和关系，同时因为本身数据来源就限定在特定领域内，知识图谱的准确性也会更高，比如微软学术知识图谱。这两种知识图谱应用范围不一样，但在本质上相差不大，更多的是数据来源的不同和筛选方式的区别。

2.1.2　知识图谱的逻辑架构

知识图谱在逻辑架构上分为两层：模式层和数据层。数据层是由许多事实组成的，上层的知识就是由一个个事实表示的。一般来说，事实是由以下几种三元组组成：关系三元组(起始实体, 关系, 目标实体)、实体属性三元组(实体, 属性,

属性值)、关系属性三元组(关系, 属性, 属性值)。另外，概念是一个模式上的结构表述，类似于面向对象中的类的概念，规定了同属于一个概念下的实体必须包含哪些属性，两个概念之间的关系可以存在哪些种类，比如人这个概念就有姓名、性别等属性，人和电影这两个概念间的关系就可以是参演、导演等。但是根据不同的知识体系组织方法，知识图谱是存在树状结构的，即与上文所述的图状结构不一样，树状结构的组织方法虽然同样可以使用三元组存储以及使用图状结构来表示（树可以看做是图的一种特殊形式，即无环的图），但在实践上更多会使用关系型数据库的数据组织方法，减少冗余，提高检索速度。

而模式层基于数据层之上构建，是表示知识的地方，通过数据层存储的事实，使用本体库进行规范化地表述知识。通过表述出来的知识更加结构化，具有更好的总结归纳和推理能力。

2.1.2 知识图谱的构建顺序

知识图谱的构建顺序分成两种：自底而上（Top-Down）和自顶而下（Down-Top）。

自顶而下的构建方式是先为整个知识图谱确定好本体库，即定义完知识图谱中存在的所有概念，这样，每一种主体所具有的属性，主体之间可能存在的关系都已经确定完毕，然后再将实体存入知识图谱。这样的构建方式更为科学和有序，因为不会出现同一种类的实体存在不同属性，在推理过程中也因为确定的结构而得到更准确的结果。但是这种构建方式需要提前分析知识的内容，一般需要借助该行业的专家的帮助，也就意味着这种构建方式更适合专业型知识图谱。

自低而上的构建方式则是在得到结构化的数据后，先添加数据，然后再在模式层总结形成本体库，这会使得实体和关系结构的不确定，在后期的知识融合和知识推理中需要考虑更多的情况，但也是无奈之举，毕竟提前预知整个本体库是非常困难的，特别是针对开放的数据，在互联网数据膨胀式增长的现在，一套通用型的本体库不能满足要求。在这种构建方式中，无需具备专业知识，而且构建本体库也可以通过自动化的方式来实施，显然更加适用于通用型知识图谱。

2.1.3 知识图谱的构建和维护

虽然知识图谱可以分成多种类型，但在大框架上是相似的，按照顺序为：信息抽取、知识融合、知识加工三个步骤，并且由于知识图谱是一个长期运行和维护的系统，需要不停地维护更新，所以在构建完毕后还需要持续不断的维护。

1） 信息抽取

结构化的知识是指使用特定的格式或语言描述知识，比如 Excel 表格中规定

每一个表用来表示什么（关系），每一列的作用是什么（属性），这样的结构化的知识才能进入下一个知识融合的步骤。而现实社会中的知识大部分都是异构的或者是半结构化的，比如通常一个新闻或网页会使用自然语言描述信息，在里面可能会存在许多形容词，语法混杂，遗失省略等问题。在如今数据量极速膨胀的大数据时代，人工标注的方法是不现实的，需要采用无监督或弱监督的方法自动化提取数据中的知识[20][21]，其中包括实体、关系和属性三个内容的提取，这属于自然语言处理（NLP）领域的研究。在本文中可以无需关心具体如何提取，对于可信知识图谱的构建，需要能够追溯知识的来源，即知识是从哪个数据中提取出来的，提取出来的结果是什么，将这两个部分的内容进行捕获即可。

表 2-1　一些现有的信息抽取算法

抽取信息	算法
实体（entity）	K-最近邻、Stanford NER 等
关系（relation）	Bootstrapping、OILLIE 等
属性（attribute）	MetaPAD、ReNoun 等

2） 知识融合

通过信息提取，可以将异构的和半结构化的数据转化为知识图谱所需的实体、关系和属性。但是显然以自动化的方法，特别是基于深度学习的算法提取的信息会存在非常多的冗余和错误，而且与现有的知识库存在歧义的问题，所以需要进行消歧后再合并[22]。实体的歧义问题可以归纳为两种：同词多义和多词同义。同词多义指同一个词可能存在多种意思，比如区块链既可以是指一种数据结构，也可以指一首歌。多词同义指同一个实体可以有多种表达方法，比如习近平总书记和习总书记，对于计算机来说，这是不同的字符串，但是实际上却是指向同一个实体。除此之外，消歧不仅针对与实体，还需要针对属性，因为属性其实也存在上述问题，比如一个日期字符串，可以使用“2022 年 1 月 1 日”表达，也可以使用“2022.1.1”表述，虽然很多编程语言的日期库能够识别出来并转化为标准形式，但也同样有许多不能涵盖的属性值，需要进行属性归一化[23]。

经过了知识融合之后，这样的知识就可以合法地插入或更新知识库，在这个步骤，涉及到的数据为步骤 1）中信息抽取得到的三元组，知识融合之后经过处理的三元组和对应的新增、更新或删除等操作类型，除此之外，同样需要注意的是，前文提到的已经结构化的数据，或者直接从已有的知识图谱数据库或关系型数据库获取到的数据，都是不需要经过信息抽取步骤的，也属于知识融合步骤的

输入数据。显然这些数据需要进行追踪，以便确实地记录下知识库中数据来源和处理方式，方便如果在数据来源变得不可信时及时发现受影响的知识。

3） 知识加工

经过信息抽取能得到原始的三元组向量，然后又经过知识融合，能减少数据的冗余和错误，自此就可以形成知识图谱中数据层里的一条条事实，但是这样的事实其实不能等同于知识，因为这样的数据太过零散，没有形成关系的网络，所以需要基于这些事实进行加工，抽象出本体库，同样也很重要的就是进行知识的推理，挖掘潜在的知识库中没表现出来的关系[24]。

如果是如上文所说的自低而上的通用型知识图谱的构建，那么本体库的构建就是在知识加工这一步骤做的，而如果是自顶而下则是需要提前到信息抽取之前完成本体库的搭建。本体是对现实世界的事物的抽象模型，更具体地说是知识图谱中的同一类实体的概念、类别，比如人、动物、国家等。主流的一般有三种方式进行组织：Ontology、Taxonomy 和 Folksonomy[25][26][27]。Ontology 是一种严格的 IsA 关系，完全的树状结构，即 B 本体是 A 本体的一种更加具体的概念，比如人与动物的关系。在这种组织方法中，实体之间的关系被严格限制，关系多样性比较低，无法覆盖现实世界中可能存在的关系，但是推理能力也极大增强，因为这些关系都是确定的，不会存在意外情况。而 Taxonomy 是在 Ontology 的基础上放宽要求，虽然仍然是 IsA 的关系，但不严格限制关系，层级关系更加低，关系多样性也更丰富。不过这样的组织方式会使得歧义增多，知识推理过程中会遇到未知的关系或潜在的关系的问题。最后的 Folksonomy 是一种标签化的组织方式，一般是使用聚类或基于已经预标注语义的知识库来生成，把标签附上到同一类实体中，这样的方式在本体库动态更新上有更大的优势，而且本体的多样性也相对丰富许多，灵活性很高，但是同时也带来关系结构的破坏的问题，不同实体间的关系难以确定，推理能力也很大程度上降低了。这三种组织方式各有优劣，应用场景也各不相同，表示形式也是不一样的，所以在处理时需要做一定区分，比如 Ontology 和 Taxonomy 是比较相似的，本体构建时会附带上概念的定义以及关系的内容，而 Folksonomy 一般只是标签的信息。

除此之外，本体库的构建虽然很大程度上是基于自动化的深度学习，比如聚类或者 Pattern-based 方法，但是仍然要人工介入来提高准确度，而这需要非常高的专业知识，知识覆盖面需要非常广，最后，这依然避免不了主观性的问题，同时也会引入安全风险，恶意攻击可以通过干涉本体库来破坏知识图谱。所以这部分人工干预的内容在本文中同样进行了追踪，捕获具体的行为。

4） 知识更新

由于现实中的事物是会不断增加和更新的，所以知识图谱的构建是一个长期

且永远不停止的过程，不能只走一次上述过程后就不再维护，而是需要通过自动化的方法不断爬取互联网上的数据（如百科类网页或社区论坛），或者由人工输入新的数据（如行业内的专业知识）。在知识图谱的构建过程中利用了大量自然语言处理技术，因此这必然会导致数据层存在着冗余和错误（因为在知识融合阶段是无法彻底消除的），甚至也包含着攻击者带来的攻击数据，这可能就需要管理员直接面向数据层进行操作，并且重新编辑逻辑层的内容，屏蔽某些恶意信息来源，这种黑名单同样也可以进行追踪，使得知识图谱不仅在数据上是安全且可溯源的，也保证知识图谱的构建程序是不可被篡改的。

2.2 区块链

2.2.1 区块链概述

区块链伴随着比特币的出现而提出，目前随着比特币的流行，其背后的区块链技术也开始备受关注，特别是区块链技术所具备的去中心化、防止篡改、数据透明公开、随时可溯源等特性，使得许多研究人员探索区块链在社会生活中解决信任问题和匿名问题。在初期，区块链广泛应用于虚拟货币，比如比特币、以太坊等方面，使用挖矿的方式发行虚拟货币，形成一个去中心化的、匿名的分布式账本。得益于虚拟货币的长时间稳定和可靠运行，区块链技术的可靠性也得到了一定证明，因此人们目前希望区块链技术不仅能用在虚拟货币上，还可以在工业、商业或者人与人交往中应用，解决目前存在的价值认证、数据溯源等问题。由此，许多企业推出了区块链综合平台，这种平台可以通过开发特定应用，同时提供电子证明可信验证、版权保护、金融安全保护等全方位的解决方案。

2.2.2 区块链的关键技术

区块链在提出时并不算是一个全新的理论，而是基于当时已有一定发展的加密哈希技术、非对称加密技术和 P2P 技术。

1） 加密哈希技术

加密哈希技术指的是通过一个函数$hash()$，使得将输入的不等长的数据 m 转化为等长的加密数据 c。

$$c = hash(m) \tag{2-1}$$

加密哈希函数有几个很重要的特征：单向性、抗第二原像攻击、抗碰撞。单向性指对于给定的已知 c，对于$c = hash(m)$无法找到对应的 m。抗第二原像攻击是指对于给定的m_1，无法找到m_2满足$hash(m_1) = hash(m_2)$。抗碰撞是指无法找到任意的m_1和m_2，使得$hash(m_1) = hash(m_2)$。

需要注意的是，这里的“无法”其实指的是在当前的计算能力上不可行，因为这样的时间复杂度或空间复杂度不是多项式级别的，目前的计算能力无法实现这样的攻击[28][29]。

但是其实这样的特征不是完备证明的，显然对于长度可能远远大于加密数据 c 的输入 m 来说，固定长度的 c 无法完全唯一表达任意的 m，所以冲突是存在的，只不过寻找这些冲突非常困难。

对于区块链来说，主要运用到的就是加密哈希函数的抗碰撞性和抗二次原像攻击，对于任何已有的数据如果恶意节点需要篡改，那必然会导致哈希值的改变，从而暴露出恶意节点的存在。哪怕由于碰撞的客观存在，对于攻击节点来说，要把现有的数据修改，并且控制哈希结果不改动，这样的代价远远大于攻击区块链所得的利益，所以也防止了恶意节点的攻击。

2） 非对称加密

与非对称加密相对，首先是对称加密。在对称加密中，加密明文使用的密钥k_1和解密密文使用的密钥k_2是相同的，在实际应用中，在非安全通道中传输的是加密后的密文，而密钥则通过安全信道传输（比如事先约定）。

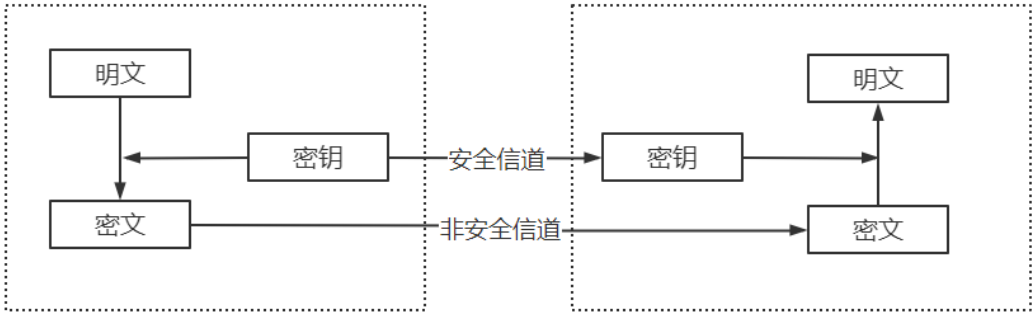

图 2-1　对称加密信息传递示意图

但是这样也存在非常多的问题，比如既然有安全信道，为什么不直接从安全信道传输明文，哪怕是事先约定密钥，也可能存在密钥泄露等问题。

而非对称加密则将加密密钥和解密密钥分开，一般解密密钥称为私钥，加密密钥称为公钥。公钥是公开的，任何人都可以获取，而私钥是保密的，由密文接受者自己妥善保管。公钥和私钥一一对应，公钥加密的内容仅私钥才能解密，并且从公钥中推到出私钥是计算困难的[30]。

典型的非对称加密 RSA 的加密方法为：

$$c = m^e \ mod \ N \qquad (2\text{-}2)$$

解密方法为：

$$m = c^d \ mod \ N \qquad (2\text{-}3)$$

在这里，m 为明文，c 为密文，e 和 N 并称为公钥，d 为私钥。

生成公钥私钥对时，需要按照以下步骤生成：

- 随机找两个质数 p 和 q，求得$N = p * q$
- 求得$p-1$和$q-1$的最小公倍数$L = lcm(p-1, q-1)$
- 在$1 < e < L$之间求出 e，使得 e 与 L 的最大公约数为 1，即$\gcd(e, L) = 1$
- 在$1 < d < L$之间求出 d，使得$e * d \ mod \ L = 1$

这样求得的公钥和私钥，符合上述的加密和解密方法，可简单证明得：

$$m = c^d \ mod \ N = (m^e \ mod \ N)^d \ mod \ N = m^{e*d \ mod \ N} \ mod \ N = m \qquad (2\text{-}4)$$

观察生成密钥对的过程可以得知，如果已知公钥$\{e, N\}$，求出私钥 d 需要分解因数 N，得到 p 和 q，然后在通过求最小公倍数、最大公因数后，得到私钥 d。根据条件可以显然发现中间变量 p、q 和 L 都是唯一的，所以只需求得 p 和 q，因为已知 e，就一定能得到私钥 d。但是由于整数分解是十分困难的，目前没有多项式级别的解法，特别是针对大素数 p 和 q，分解所需的时间是无法接受的，所以我们可以认为公钥无法推导出私钥，从而证明了非对称加密的安全性。而事实上，非对称加密于 1976 年提出，经过了数十年的发展，实践也证明了其安全性和可靠性。

在区块链中，非对称加密用于身份识别和账户信息的加密。区块链中的数据都是透明的，所有交易任何人都可以访问到，但是对于其他人来说，仅能知道是某个账户参与了交易，但并无法知道这个账户的信息，也就无法知道这个账户属于现实中的哪个人的。通过非对称加密算法，区块链实现了匿名化，即将现实中的人的信息与区块链中的账户的信息隔绝来开，其他人无法通过区块链来追踪账户的所有者。

3） P2P

区块链是个去中心化的系统，由于不存在服务器，每个节点都是平等的，即属于服务器又属于客户端。但同时也就带来了寻找节点的困难，在比特币中，对于一个希望加入区块链网络的节点，需要先使用洪泛（flooding）技术[31]，获取周围的节点，然后通过周围的节点，再扩充节点对区块链网络的发现。在数据交互的过程中存在着不同的方式，比如比特币网络使用全分布式非结构化 P2P 网络，所有数据需要通过相邻节点转发，而以太坊使用全分布式结构化 P2P 网络，可以直接得知需要的资源的节点信息。通过 P2P 技术，区块链网络得以实现去中心化，各节点无需中心节点直接互相通信，部分节点受到攻击也不影响整个网络的运行。

2.2.3 区块链的结构

区块链是以一个链式结构构成的，链上的每一个单元称作区块，区块之间使用哈希值连接。

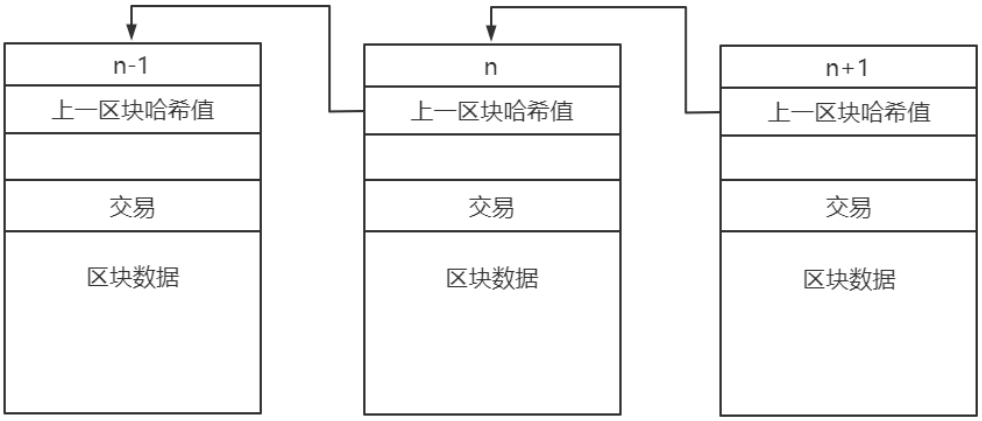

图 2-2 区块链的区块以哈希值连接

区块的内容中是一系列的交易，交易也是区块链中的重要主体，转账、智能合约的操作等都属于交易的内容，当生成区块时，节点会打包当前存在的还未记录进区块链之前区块中的交易，放入区块的区块体中。

通过当前区块打包的交易，生成一个 Merkle 树[32]，这个树也是比特币的区块体中的数据。如图 2-3 为 Merkle 树的结构。

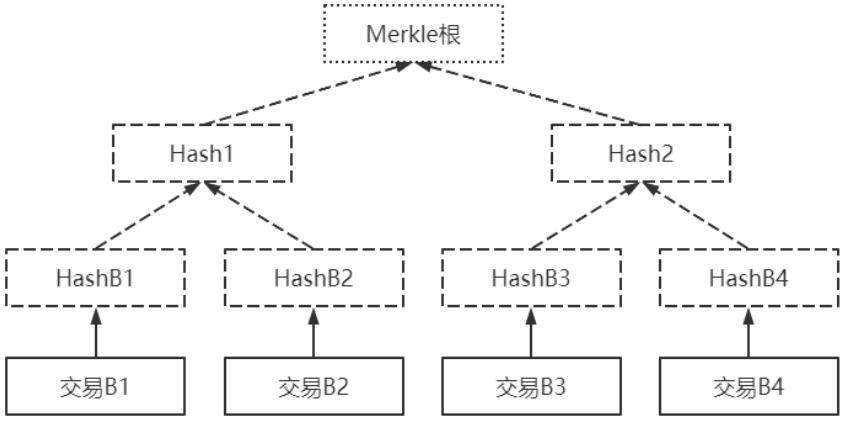

图 2-3 Merkle 树

Merkle 树的叶子节点都是区块中的交易，每一笔交易都生成一个哈希值，向上形成根节点时，则是两个哈希值连接然后再求哈希值，如此递归向上直到求

出 Merkle 数的根节点哈希值，称为 Merkle 根，并将其放置在区块的首部。

$$HashB1 = Hash(B1) \quad (2\text{-}5)$$

$$HashB2 = Hash(B2) \quad (2\text{-}6)$$

$$Hash1 = Hash(HashB1 \mid HashB2) \quad (2\text{-}7)$$

除此之外，区块的首部同样还有前一区块的哈希值，也就是通过这个字段使得区块链称为一个链状结构联系起来。当恶意节点尝试修改已有的区块中的交易时，就会导致 Merkle 根的变动，同时也就导致整个区块的哈希值变动，使得区块链无法连接起来，从而使其他节点发现恶意行为，阻止恶意节点的攻击。

2.2.3 区块链的共识机制

区块链的共识机制是在去中心化的各个节点权利平等的网络中，协调对区块数据的验证和产生的共识问题[33]。比较常用的有以下几种：

1）工作量证明（Proof of Work，PoW）。任何人都能生成区块，但是生成的区块必须要符合一定的条件，比如这个区块在填入了正确的交易信息、上一区块哈希值等数据后，还需要生成一个随机数，使得这个新区块的哈希值满足一定要求。由于哈希算法无法通过哈希值推测原始数据，即单向性，所以这个随机数的生成几乎只能穷举，在要求严格的情况下，穷举的次数会非常庞大。这样带来的好处是在一个庞大的区块链网络中，少数恶意节点的算力远远比不上整个网络的算力，攻击网络成为一个非常困难的事。缺点也很明显，即会带来大量的资源浪费。通常虚拟货币使用的都是 PoW 的方式运行，并且通过生成区块的方式发行货币。

2）权益证明（Proof of Stake，PoS）。任何人都能生成区块，并且生成区块的节点是随机产生的，不严格要求新区块的哈希值。但是随机选择节点是有偏向性的，当节点拥有的资产越多，持续时间越长，被选中的概率就越大。这样的共识机制无需大量的算力即可运行，极大减轻了资源的浪费，但是同样也会导致“贫富差距”的增大，节点的资产越多，生成区块的可能性越大，获得奖励就越多，从而资产增加，如此往复，富有的节点会更加富有。目前，为了解决虚拟货币带来的资源浪费问题，有些虚拟货币计划向 PoS 转移，比如以太坊。

3）委托权益证明（Delegated Proof of Stake，DPoS）。不是所有人都能产生区块，需要在网络中选取特定的某些节点，按照一定顺序轮流生成区块。引入了选举的功能，区块链中的节点可以投票给某个节点，在每一轮循环中，区块链会自动选择最多票的 n 个节点，然后随机生成一个顺序，这 n 个节点则按照顺序依次产生区块。在这种共识机制下，节点需要关注被选举的节点的工作状态，如果其是恶意节点或者状态异常，则可以将其排除出去。DPoS 解决了 PoS 中不可避

免的贫富差距拉大的问题，只要区块链中的节点保持活性，那么就能维持高效的运作。

2.2.4 区块链的智能合约

智能合约简单来说就是电子合同，但是是电子合同的一种新表现[34]。合同是了双方之间设立、变更、终止民事关系的协议，在这个意义上，智能合约与合同无异，买卖合同变为智能合约的买卖合同，原本以纸质形式表述的合同变成使用代码形式表述的约定。但是也有不一样的地方，传统合同的实施先要靠双方的自觉性执行，如果有与合同存在出入的地方，再确定是否提出诉讼请求，最后再由法院判定结果。但在智能合约上，合同的约束不仅表述在代码上，连合同的执行过程一样表述在代码中，如买卖合同，买方提前把钱存入智能合约，在收到货后，会自动把钱转入卖方账号，在这个过程中不会被外来因素干涉。除此之外，现实中的合同存在伪造和污损的可能性，而智能合约作为区块链上的数据的一部分，天然具有防止篡改的功能。

在区块链中，智能合约可以用来做很多事，它不仅仅只是一个合同，也可以基于这样的安全性开发各种应用，比如提供稻米供应链动态监督模型、电力交易模型，甚至是开发小游戏。在本文中，对知识图谱的可信构建和溯源也是基于构建智能合约来实现的，将知识图谱的构建过程产生的数据和操作日志进行上链处理，确保所有操作包括恶意行为都公开透明地记录下来，同时提供相应的检索功能。

2.3 本章小结

本章详细地论述并概括了知识图谱所有主流形式构建的过程和形式，分析了每一个步骤的作用和涉及的数据信息，并初步总结本文需要关注的内容，为下文建立一个通用的知识图谱构建模型奠定理论基础。并且分析了区块链的核心技术和功能，对其工作机制和安全性原理有了更深入的理解，初步确定了本文需要对区块链研究的关键点。

第三章　面向知识图谱构建的区块链溯源

3.1　知识图谱构建溯源模型

本文提出的面向知识图谱构建的区块链溯源模型，是为了解决在传统知识图谱的构建过程中，由于经过了自动化（如基于 NLP 的信息抽取、知识融合等）和人工介入的评审和干涉（如本体库的构建、数据纠错等），知识图谱中的数据来源不明，无法追溯错误数据的数据源或人工审核过程，或者在确定污损的数据源后，无法便捷地追溯这些数据源影响的知识库数据，导致不能很好地对数据源和构建过程进行定位错误和修复更新。由于很多现有的知识图谱的应用场景和开放程度不同，比如百度和谷歌的知识图谱平台都允许用户自行添加数据，但是百度的知识图谱平台可以直接通过 API 进行上传数据，而谷歌的知识图谱平台更倾向于用户在知名平台如维基百科、LinkedIn 填充数据，然后平台会自动去获取数据，而微软学术知识图谱几乎不开放，仅自己去爬取数据。同时，与上文描述的一致，虽然大多数知识图谱在数据层是相同的，但在逻辑层和具体的核心代码有非常大的不同。而且，知识图谱系统是一个已经存在并作为一个重要产品来使用，本文新引入的可信构建和溯源方法希望能在一定程度上尽量避免对现有系统的影响，同样，区块链的引入不仅可以从头构建，也可以与现有的区块链平台结合，使之更加具有可靠性。因此，本文提出的面向知识图谱构建的区块链溯源模型可以分为以下四个耦合性较低的模块。

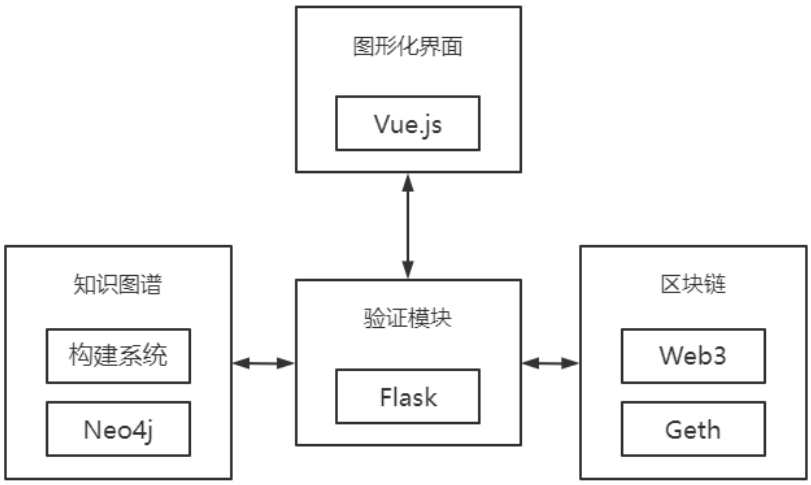

图 3-1　整体架构图

3.2 通用型知识图谱

本文提出的知识图谱可信构建和溯源方法是一个通用的方法，基于分析知识图谱在构建过程的相似之处和区块链的可扩展性，这里提出一种通用的知识图谱构建模型，如图 3-2，本文的工作将基于此模型进行。

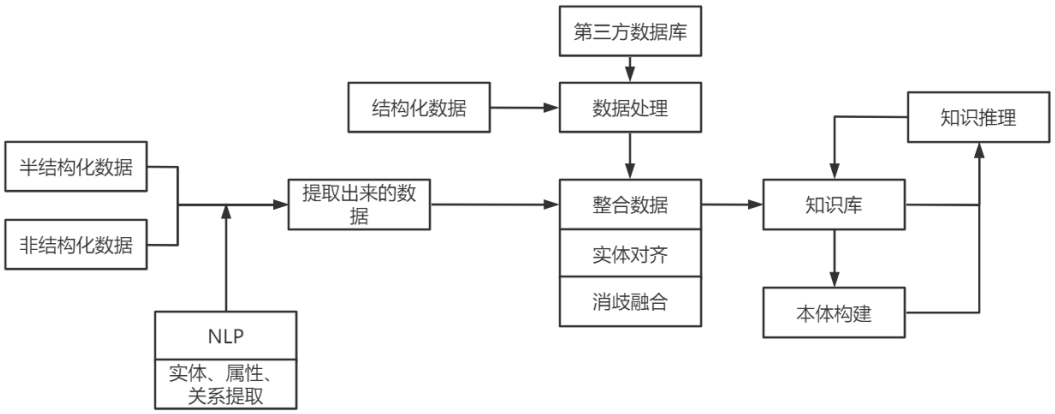

图 3-2 通用知识图谱构建模型

3.2.1 信息抽取

在信息抽取过程中，输入的数据为采集的半结构化或非结构化的数据，经过实体抽取、属性抽取、关系抽取，可以得到结构化的关系，形成三元组向量。

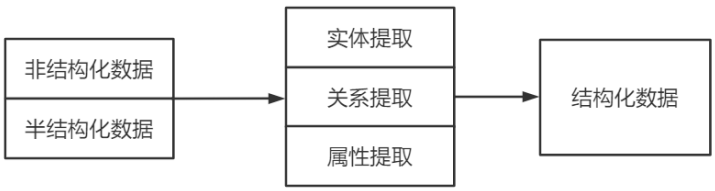

图 3-3 信息抽取流程图

这里由于实体提取、属性提取和关系提取一般是应用不同的算法，并且每种内容的抽取算法也有多种选择，所以结果格式各不相同，所以需要统一数据结构。

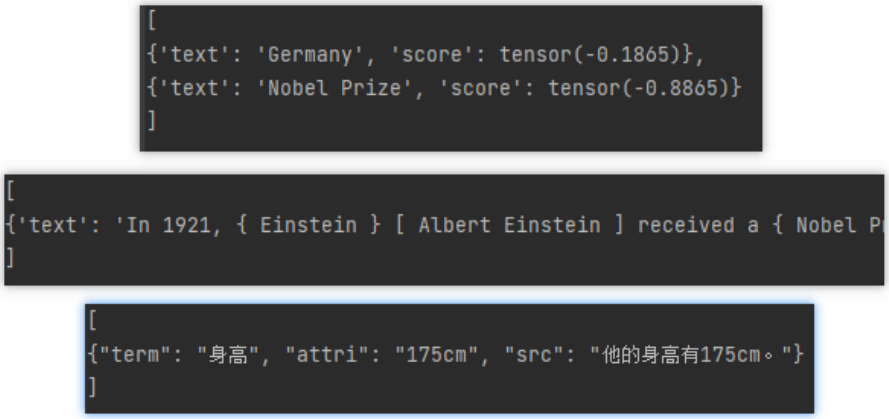

图 3-4　分别是实体、关系、属性的抽取结果示例，不同的算法有不同的输出。

名称	类型	必填	默认值	描述
src	string	true		来源语句
score	string	true		置信度
result	array	true		
text	string	true		row result
head	string	true		起始
tail	string	true		结束
relate	string	true		关系名

图 3-5　信息抽取记录数据结构

这样设计的数据结构可以涵盖大部分信息抽取得到的结果，并将这样的数据记录到区块链中。

3.2.2　知识融合

在知识融合阶段，主要是为了将数据中的实体对齐、消歧，在这里的数据来源不仅有信息抽取中的数据，同样还有第三方数据库和已经结构化的数据，这部分数据需要事先进行数据整合。

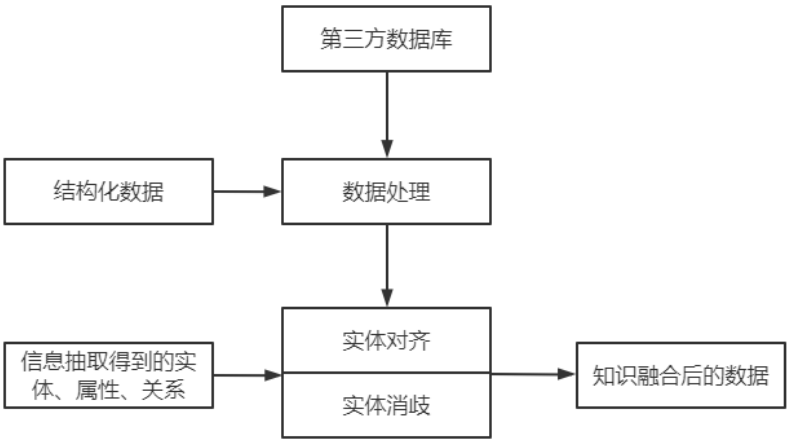

图 3-6　知识融合流程图

3.2.3　知识加工

在知识加工阶段，会进行本体的构建，根据不同的知识组织方式，会产生不一样的本体库结构，同时由于语义复杂，虽然通过人工编辑的方式手动构建不太现实，但是在使用机器学习算法构建后，再使用人工审核的方法是很有必要的。

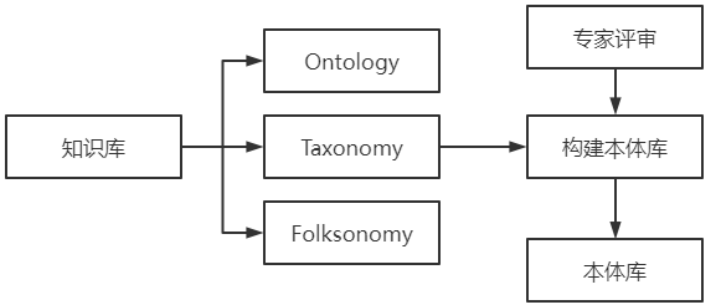

图 3-7　知识加工流程图

3.2.4　知识推理

在知识推理阶段，基于已有的知识库基础上进一步寻找实体间暗藏的知识，从而丰富、扩展知识库。知识推理可以是基于逻辑的推理，也可以是基于图的推理，两种推理方式依据不同，前者属于通过本体的概念层次进行推理，更加倾向于实体的语义关系，而后者是依据图上的路径进行推理，根据路径预测它们的关系，这种推理方式使用较少。

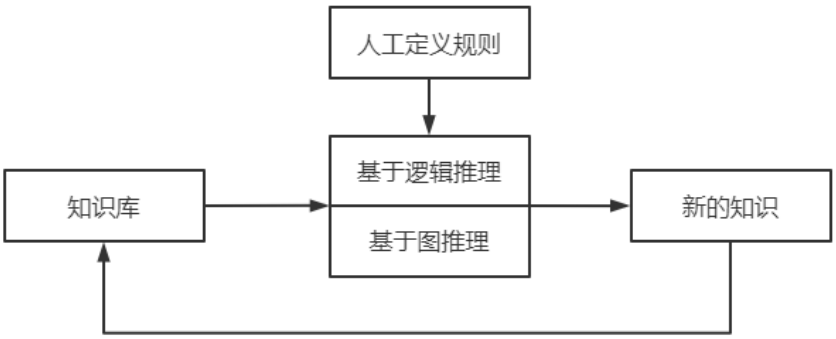

图 3-8　知识推理流程图

并且知识推理需要比较大的人工预定义和审核工作，人在其中的参与度较高，所以也需要对人工的数据进行捕获。

3.3　区块链溯源

本文提出的知识图谱的可信构建和溯源的可信是基于区块链的安全性，区块链中不同的共识机制、结构等有所差异，需要根据需求选择，本文提出的方法需要满足在一个开放型的知识图谱中进行，任何人都可以参与维护和更新知识图谱（如百度知识图谱平台），所以要尽可能地降低资源消耗，提高效率。

3.3.1　共识机制的选择

比对不同的共识机制，由于本文的方法不需要使用工作量证明，而且 POS 的缺陷可能导致中心化，所以本文将采用 DPoS 的共识机制运行。在这种共识机制下，只有被选择出来的超级节点才能生成区块，基于投票选择的方式选举可以避免过度中心化。在每一轮选举中，超级节点的产生区块的顺序是随机选定的，而每个超级节点只有在自己的那一部分时间内才能生成区块，如果不生成新的区块，则会被跳过，即不获得收益，这样的行为会导致区块链网络的低效运行，自然也会被其他节点舍弃。在本文中，除了基于原本的 DPoS 共识机制，同时也提出了可自由定义投票收益比例，这样的机制使得投票的节点跟随着超级节点的运行而获益，从而提高整个网络的活跃度。

3.3.2　智能合约的设计

智能合约除了原本的功能外，主要用来记录上文提到的知识图谱构建过程中的数据。本文设计了一个共识机制，按照 3.2 中的数据格式提供系统的数据上传和检索，通过检索函数，可以根据需要获取任何一个步骤中的数据来源和去向。

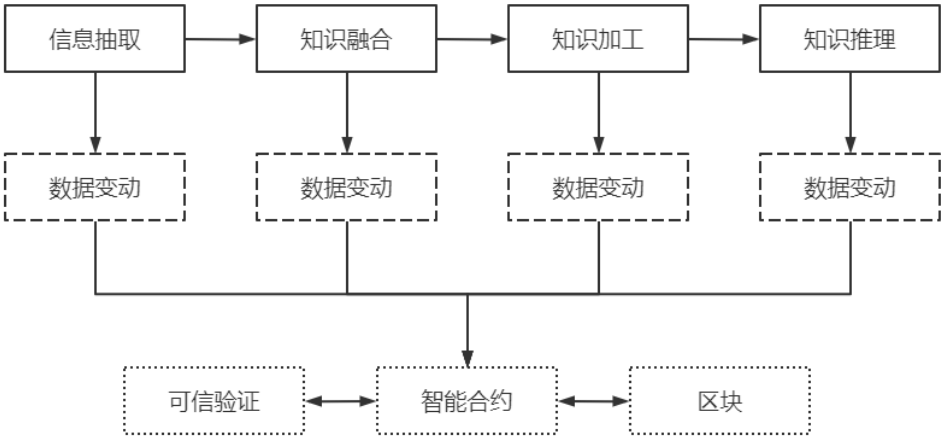

图 3-9　通过智能合约收集知识图谱构建过程中的信息，并提供数据的可信验证功能

3.4　本章小结

第三章详细地阐述了本文提出的知识图谱可信构建和溯源模型，设计了一个通用型的知识图谱构建模型，尽可能覆盖目前主流的知识图谱架构，并描述具体每一步出现的信息转化的流程。在区块链的设计中，采取资源消耗更低、效率更高的 DPoS 共识机制，并针对知识图谱构建模型中的每一个步骤都设计对应的共识机制接口，对接构建中产生的信息。最后再通过验证模块，根据区块链上的数据，自动串联数据源、中间数据、知识，提供对知识的溯源和定位数据源产生的知识的功能。

第四章　实验及结果分析

4.1　实验环境介绍

本实验所用设备为个人电脑，并通过创建多个虚拟机模拟多个节点同时通信。虚拟机上分别运行不同的区块链客户端、知识图谱程序。

个人电脑的配置如下：操作系统为 Windows10 专业版，处理器为 Intel i5 8400，6 核心 6 线程。内存 16GB。GPU 显卡为 GTX 1050Ti，显卡内存 4GB。

本次实验终端模块/系统使用了多个开发平台，区块链系统使用基于 Go 的 Geth 以太坊客户端，并使用基于 Solidity 的 Remix 编写智能合约；在知识图谱系统端，使用 Neo4j 做数据持续性存储，并使用 Python 编写上文描述的知识图谱构建模型；在验证模块使用前后端设计，前端使用 Vue.js 框架构建知识图谱的编辑和验证的可视化界面，后端使用 Flask 框架提供接口支持。

表 4-1　个人笔记本电脑实验环境配置

软硬件条件	参数
操作系统	Windows 10 专业版 21H1
CPU	Intel i5 8400 2.8GHz 6 核心 6 线程
GPU0	Intel UHD Graphics 630
GPU1	NVIDIA GeForce GTX 1050 Ti
GPU2	NVIDIA GeForce GTX 980 Ti
RAM	16GB
VMware	16.2.2 Pro

表 4-2　虚拟机配置

软硬件条件	参数
操作系统	Ubuntu 18.04 LTS
CPU	Intel i5 8400 2.8GHz 4 核心
RAM	4GB
磁盘	40GB
网络	NAT 类型，连通互联网

表 4-3　区块链系统环境配置

环境	参数
GoLang	1.10.4 linux/amd64
Geth1	1.10.17-stable based on go1.18
Geth2	1.7.4-stable based on go1.10.4
JetBrains GoLand	2022.1.1 Professional
web3	4.10.0
solidity	0.4
Remix	0.23.3

表 4-4　知识图谱系统环境配置

环境	参数
Neo4j	Neo4j 4.4.4
JetBrains Pycharm	2021.3.1 Professional
Python	3.7.9
py2neo	2021.2.3
pandas	1.3.5

表 4-5　验证模块系统环境配置

环境	参数
Vue.js	2.6.14
Vue-cli	5.0.4
ElementUI	2.15.8
Echarts	5.3.2
HBuilder X	3.3.13.20220314
Flask	2.1.2
JetBrains Pycharm	2021.3.1 Professional

4.2　实验数据

本次实验的数据来源于网上的开源数据库，是一个关于公司、股东、个人、股票等信息的知识图谱，由于本文并不需要关注信息抽取、知识融合、知识加工过程中使用 NLP 技术处理的具体过程，只需要记录每个过程中数据的变化即可，

所以将采用已经结构化的数据来模拟半结构化和非结构化数据，验证本文提出的方法的正确性。总共 10 个 csv 文件，总计 99510 项数据，其中包含一些冗余字段，可供知识融合时使用。

图 4-1　部分数据集信息

4.3　实验内容及结果分析

4.3.1　系统实现

1）知识图谱部分

知识图谱的数据层使用 Neo4j 图数据库提供持续性存储，这个数据库是知名开源数据库，支持图类型的数据存储，并且内置图形化界面，使用 Cypher 语句对数据库进行操作。

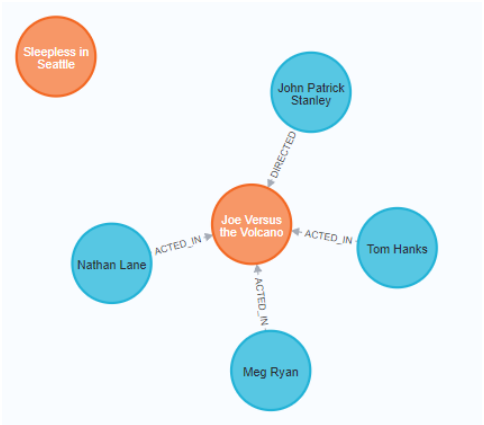

图 4-2　Neo4j 数据库图形化显示

Neo4j 的数据类型存在两种：节点和边。每个节点和边都可以存在任意多个属性和属性值对；每个节点可以存在多个 Label，用以区分不同类型的节点；边没有 Label，只有一个边名。在这里，每一个 Label 对应着一个知识图谱的本体，属性和属性值对为知识图谱中的实体和关系的属性，边名即关系名。

知识图谱的构建过程使用 Python 实现，需要先封装 Neo4j 数据库工具类，方便后续在知识图谱的构建中对数据库的操作。工具类提供了对数据库的节点和边的增删改查操作。

```
class Neo4jHelper:
    def __init__(self):
        self._hd = Graph(neo4j_url, auth=(neo4j_username, neo4j_password))

    def add_node(self,
                 labels: tuple = (),
                 properties: dict = {}):...

    def add_node_all(self,
                     nodes: list = []):...

    def del_node(self, node_id):...

    def get_all_node(self,
```

图 4-3　Neo4j 数据库工具类部分核心代码

知识图谱的构建使用面向对象的方法，每一个数据源都需要完整地走一次信息提取和知识融合过程，形成为三元组向量，然后再转化为 Neo4j 的格式写入数据库。

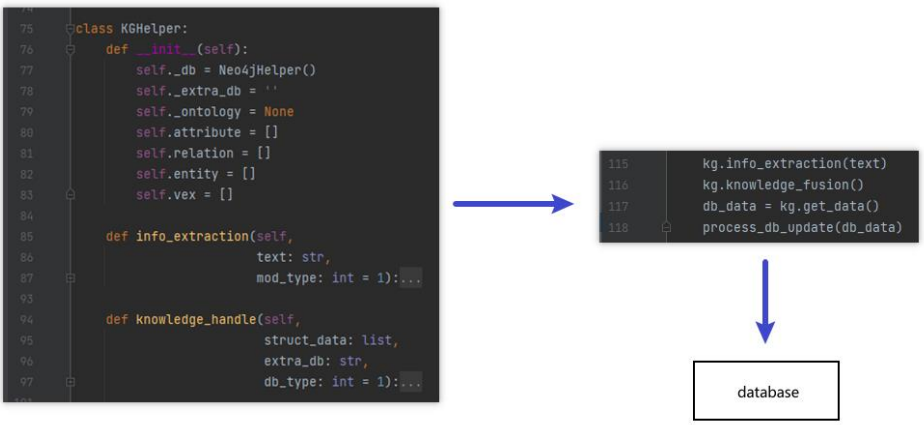

图 4-4　知识图谱构建模型的实现部分核心代码

2）区块链部分

本文的实验中使用的区块链系统是基于 Geth 以太坊的改造而来，原版以太坊是使用 PoW 的共识机制，但本文提出的方法为了避免资源浪费，需要改编为 DPoS 共识机制。Geth 是基于 Go 语言编写的开源区块链，修改源代码即可实现共识机制的修改，并且 Github 等开源平台上也有许多相似的分支。在本文中，通过修改 mine.go 和 validate 模块，使得其实现选举、产生区块的功能。

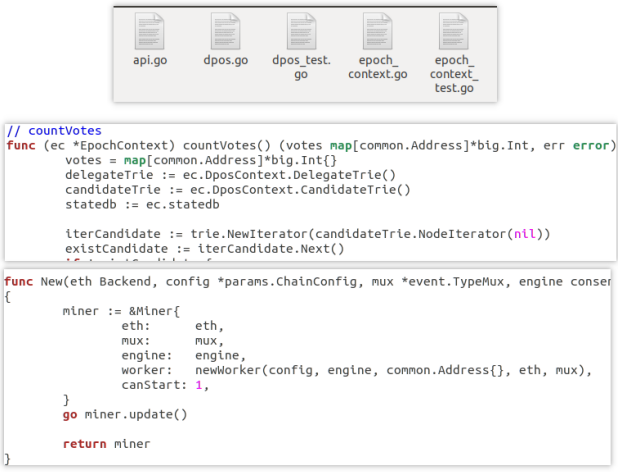

图 4-5　区块链 DPoS 共识机制相关关键文件和核心代码

由于 DPoS 需要选举超级节点生成区块，本文中为了方便验证，使用 5 个超级节点，所以需要建立 5 个区块链客户端，互联共同维护区块链的运行。

在智能合约部分，本文使用 Remix 在线编译器，基于 Solidity 语言，编写了一个智能合约，用于对知识图谱构建过程产生的信息记录，并提供相应的查找功能。

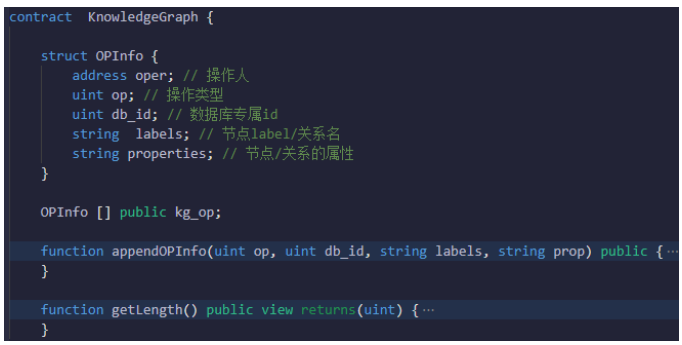

图 4-6　智能合约部分代码

最后，对接区块链使用了 Python 的 web3 包，并基于此编写了一个工具类，该工具类提供对 Geth 的连接、区块链的运行、智能合约的操作等功能。由于官

方 Geth 尚未迁移到 PoS 共识机制，所以对应的 web3 包存在一些问题，需要更改，主要是修改中间件，使之支持 DPoS 的区块数据格式，以及对较低版本的兼容性。

```python
class GethHelper:
    def __init__(self):
        self._w3 = Web3(Web3.HTTPProvider(geth_url))
        self.init_web3()

        # init contract
        address = self._w3.toChecksumAddress(geth_address)
        self._con = self._w3.eth.contract(address, abi=geth_abi)

    def init_web3(self):
        # set dpos
        self._w3.middleware_stack.inject(geth_poa_middleware, layer=0)

        # set default account
        if len(self._w3.eth.accounts) > 0:
```

图 4-7　geth 工具类，提供区块链的对接功能

3）验证模块部分

验证模块处于区块链和知识图谱之间，提供可视化的知识图谱数据导入、验证和区块链的操作。模块使用 Browser/Server 架构，前端基于 Vue.js 编写，使用 ElementUI 和 Echarts 提供美化的界面和数据显示，后端基于 Python 的 Flask 框架，与区块链部分和知识图谱部分对接，并以 HTTP 接口的形式为前端提供数据支持。

图 4-8　左图为前端查询链上数据，右图为后端程序文件结构

4.3.2　实验结果和分析

1）完备性验证

通过前端界面导入数据集的 10 个 csv 文件，以此作为数据源，通过本文提出的知识图谱构建模型，生成特定的格式存入数据库，与此同时，对知识图谱构建过程产生的数据传入区块链。

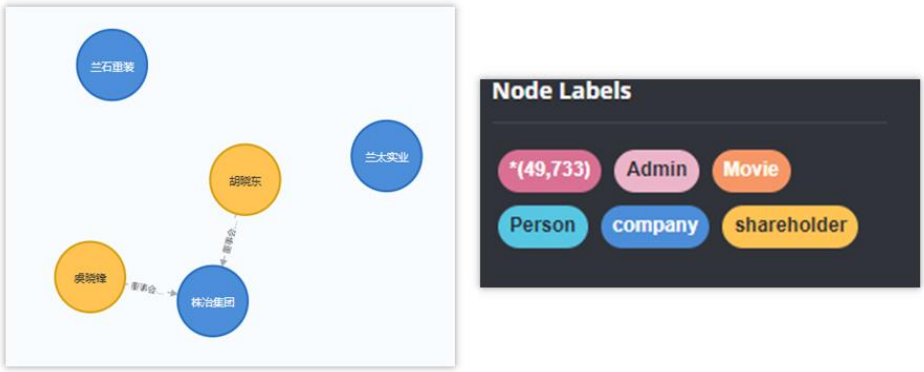

图 4-9　知识图谱部分信息

0x1A753620556C6aA8f5fE4830117E8Ad17C4E7253	添加节点	188	Person, shareholder	name: 虞晓锋
0x1A753620556C6aA8f5fE4830117E8Ad17C4E7253	添加节点	189	Person, shareholder	name: 胡晓东
0x1A753620556C6aA8f5fE4830117E8Ad17C4E7253	添加节点	190	company	name: 株冶集团
0x1A753620556C6aA8f5fE4830117E8Ad17C4E7253	添加关系	260	董事会成员	

图 4-10　对应区块链上的数据

可以看到，数据源中的数据已经转化为事实存入数据库中，通过 Neo4j 和 Echarts 模块，可以快速查询到事实对应的数据库信息，并且可以通过验证模块获取到区块链上的区块信息和通用知识图谱构建过程中知识的转化过程。由此，基于区块链的去中心化和不可篡改的特性，系统提供了一个可供多节点同时运行的知识图谱安全构建功能，并且可以借助智能合约中的接口使用验证模块追溯知识图谱中的任何数据的来源、去向，解决了传统知识图谱中存在的知识溯源、鉴权问题。

2）性能对比

本文提出的方法在资源消耗上比一般区块链低许多，因为无需花费计算资源来决定哪个节点能产生区块，而是通过投票选举的方式选出超级节点来轮流生成区块。本文在做实验时，因为 DPoS 和 PoW 两个共识机制在 Geth 上都有实现，

所以几乎可以直接对接本系统，经过实际运行，可以得到下图的对比结果。

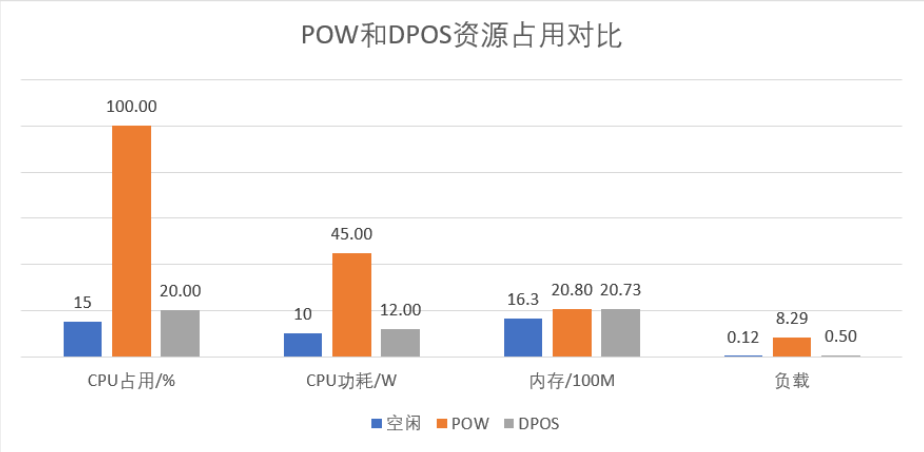

图 4-11　对应区块链上的数据

如图 4-11，可以发现基于两个共识机制的区块链在内存上相差不大，但在 CPU 资源的消耗上相差巨大，这是由于 PoW 需要持续地计算哈希值导致的。值得注意的是，PoW 的挖矿行为一般在使用 GPU 时会获得更高性能，这样可以解放 CPU 来处理其他任务，但不管如何，这样的资源浪费是需要尽量避免的。

除了资源消耗之外，本文还关注区块链的运行效率问题，在实验中，通过持续性快速生成交易，并检查剩余交易量，可以计算出大概平均每秒处理的交易量。如表 4-6 所示，相对于比特币和以太坊，处理效率有一定提升，这是因为本文的区块结构稍微有所更改，扩大了区块的大小，而且通过控制每个超级节点生成区块的时间，可以成倍提高处理效率。

表 4-6　不同区块链处理交易效率对比

区块链	效率
比特币	7 笔/秒
以太坊（PoW）	85 笔/秒
以太坊（DPoS，3s）	350 笔/秒
以太坊（DPoS，1s）	900 笔/秒

第五章　总结与展望

5.1　总结

知识图谱在当今计算机领域有着非常重要的作用，它提供了一种计算机可以使用的关系检索和推理能力，使得内容多源、数据多样的互联网信息得到充分地组织，转变为计算机能处理的知识。因此，知识图谱广泛应用于实现语义化智能检索和知识互联等领域，比如搜索引擎、推荐系统。

但因为知识图谱需要的数据十分庞大，知识图谱平台往往需要自动化爬取互联网上的数据，或者开放编辑权限给其他用户来更大范围地获取知识。虽然一般知识图谱平台都会存在审核人员把控数据的准确性，但局限于审核人员的专业性和数据量的庞大，完全抵抗恶意攻击是十分困难的。并且在知识的抽取中，需要应用到 NLP（自然语言处理）技术，高度的自动化同时也导致知识库中的数据来源难以追溯，在已知恶意数据源时难以定位被污染的知识，在确定错误知识时难以追溯缺陷数据源。

而区块链技术的出现，基于其去中心化、防篡改的特点，可以为解决现实中存在的中心化、数据不安全、可信交易等问题。虽然区块链所基于的密码学原理是依靠目前计算资源的计算困难，但多年来虚拟货币的可靠运行和其他哈希算法和非对称加密算法的应用，可以充分证明区块链技术的可靠性和安全性。因此，区块链技术已经成为当今解决计算机领域内解决安全和信任问题的重要方案。

因此，本文创新性地提出使用区块链技术，解决知识图谱在开放性的构建过程中遇到的知识溯源和可信验证困难的问题。首先详细地分析了目前主流的知识图谱架构和原理，利用主流的知识图谱构建过程提出了一套通用型的知识图谱构建模型，使得本文提出的方法更具兼容性。然后考虑到知识图谱应用场景的特殊性，不能消耗过多平台和普通用户的计算资源，为了更加耦合知识图谱的应用，本文利用现有的以太坊 Geth 区块链修改共识机制和编写智能合约，使得系统在资源消耗上有很大的降低，并且处理效率也有了极大提升。最后，通过验证模块对知识图谱构建的信息进行查询与追溯，实现知识图谱构建数据的溯源。

本文通过实验验证了提出的知识图谱构建模型和提出的验证方法的正确性和可靠性，并且根据对比，显示了本方法在资源消耗上和处理效率上都有一定提升。

5.2 展望

本文提出的是面向知识图谱构建的区块链溯源方法研究，为了兼容性，提出了一套通用的知识图谱构建模型，但仍然不可能覆盖全部的知识图谱架构，比如现有的树状的知识图谱，在未来，也有可能会出现更多的更高效的知识图谱模型，所以，对于知识图谱的可信构建和溯源还有更多需要研究的地方。

除此之外，本文使用的区块链是基于已有的以太坊改编的，但原生以太坊是使用 PoW 的共识机制，许多内部方法和结构也依赖于此，所以在修改为 DPoS 后，内部存在着许多无用字段和性能消耗，而且外部的工具和文档支持也将有所出入，处理效率进一步提高也存在着困难。因此，在未来的研究中，可以从头设计全新的基于 DPoS 的区块链系统，改善区块结构和运行方法，以求取得更加高效的运行效率。当然，由于目前区块链技术的发展，许多区块链平台也在形成，利用现有的区块链平台来为本方法提供支持也是一个很好的选择。

参考文献

[1] AMIT S. Introducing the knowledge graph[R]. America: Official Blog of Google, 2012.

[2] 王浩. 大数据时代个人隐私保护意识崛起研究 ——基于 CiteSpace 的可视化知识图谱分析[J]. 河南科技,2021,40(19):11-14.
DOI:10.3969/j.issn.1003-5168.2021.19.015.

[3] LIU Yu, LI Yang, DUAN Hong, et al. Knowledge graph construction techniques[J]. Journal of computer research and development, 2016, 53(3): 582 - 600.

[4] 常亮,张伟涛,古天龙,孙文平,宾辰忠.知识图谱的推荐系统综述[J].智能系统学报,2019,14(02):207-216.

[5] ZHONG Cui-jiao. Research on semantic organization of web information and retrieval[J]. Research on Library Science, 2010, 75(17): 68-71.

[6] 徐增林,盛泳潘,贺丽荣,等. 知识图谱技术综述[J]. 电子科技大学学报,2016,45(4):589-606. DOI:10.3969/j.issn.1001-0548.2016.04.012.

[7] Nakamoto S. Bitcoin： A peer-to-peer electronic cash system. White Paper, 2008.

[8] Li X，Jiang P, Chen T，et al. A survey on the security of blockchain systems. Future Generation Computer Systems, 2017, 8： 274.

[9] Diffie W, Hellman M. New directions in cryptography. IEEE Transactions on Information Theory, 1976，22(6): 644-654.

[10] BIZER C, Al E. Linked data-the story so far[J]. International Journal on Semantic Web & Information System, 2009, 5(3): 1-22.

[11] Amit Singhal. Introducing the Knowledge Graph: things, not strings. [2012-03-16].
https://www.blog.google/products/search/introducing-knowledge-graph-things-not/.

[12] 王硕,杜志娟,孟小峰. 大规模知识图谱补全技术的研究进展[J]. 中国科学（信息科学）,2020,50(4):551-575.

[13] 张美璟. 知识图谱在犯罪情报分析中的应用[J]. 法制与社会,2021(4):82-83,112. DOI:10.19387/j.cnki.1009-0592.2021.02.038.

[14] 蔡晓晴,邓尧,张亮,等. 区块链原理及其核心技术[J]. 计算机学报,2021,44(1):84-131. DOI:10.11897/SP.J.1016.2021.00084.

[15] Gerstl D S. Leveraging Bitcoin blockchain technology to modernize security perfection under the uniform commercial code//Proceedings of the International Conference of Software Business. Ljubljana, Slovenia, 2016: 109-123.

[16] He Pu，Yu Ge，Zhang Yan-Feng, et al. Survey on blockchain technology and its application prospect. Computer Science, 2017, 44(4)： 1-7.

[17] 丁晓蔚,何秋妍. 论区块链技术对传媒业的影响[J]. 现代传播,2019(12):9-13,20. DOI:10.3969/j.issn.1007-8770.2019.12.003.

[18] WALCZAK S.. Knowledge-based search in competitive domains[J]. IEEE Transactions on Knowledge and Data Engineering,2003,15(3):734-743.

[19] LI, L., LI, X., CHENG, C., et al. Research Collaboration and ITS Topic Evolution: 10 Years at T-ITS[J]. IEEE transactions on intelligent transportation systems,2010,11(3):517-523.

[20] 孙镇, 王惠临. 命名实体识别研究进展综述[J]. 现代图书情报技术, 2010(6): 42-47.

[21] ZHAO Jun, LIU Kang, ZHOU Guang-you, et al. Open information extraction[J]. Journal of Chinese Information Processing, 2011, 25(6): 98-110.

[22] SOCHER R, CHEN D, MANNING C D, et al. Reasoning with neural tensor networks for knowledge base completion [C]//Proc of NIPS. Cambridge, MA: MIT Press, 2013: 926-934.

[23] 苏佳林,王元卓,靳小龙,等. 融合语义和结构信息的知识图谱实体对齐[J]. 山西大学学报（自然科学版）,2019,42(1):23-30. DOI:10.13451/j.cnki.shanxi.univ(nat.sci.).2018.11.05.003.

[24] WONG W, LIU Wei, BENNAMOUN M. Ontology learning from text: a look back and into the future[J]. ACM Computing Surveys, 2012, 44(4): 18-24.

[25] 张成洪,张杰伟. 基于Ontology的跨组织知识整合方法研究[J]. 管理工程学报,2011,25(3):53-61. DOI:10.3969/j.issn.1004-6062.2011.03.010.

[26] WU Wen-tao, LI Hong-song, WANG Hai-xun, et al. Probase: a probabilistic taxonomy for text understanding [C]//Proc of the 31st ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2012.

[27] 薛涵,秦兵,刘挺. 基于 Folksonomy 的本体构建综述[J]. 电子学报,2014(4):791-797. DOI:10.3969/j.issn.0372-2112.2014.04.026.

[28] 李志敏. 哈希函数设计与分析[D]. 北京:北京邮电大学,2009.

[29] 郭一村,陈华辉. 在线哈希算法研究综述[J]. 计算机应用,2021,41(4):1106-1112. DOI:10.11772/j.issn.1001-9081.2020071047.

[30] 王宏杰. RSA 算法、DES 算法的特点分析及结合[J]. 天津科技,2005,32(4):37-38. DOI:10.3969/j.issn.1006-8945.2005.04.014.

[31] 武岳,李军祥. 区块链 P2P 网络协议演进过程[J]. 计算机应用研究,2019,36(10):2881-2886,2929. DOI:10.19734/j.issn.1001-3695.2018.07.0365.

[32] 刘竹松,何喆. 基于Merkle哈希树的云存储加密数据去重复研究[J]. 计算机工程与应用,2018,54(5):85-90,121. DOI:10.3778/j.issn.1002-8331.1610-0038.

[33] 谭敏生,杨杰,丁琳,等. 区块链共识机制综述[J]. 计算机工程,2020,46(12):1-11. DOI:10.19678/j.issn.1000-3428.0059070.

[34] 郎芳. 区块链技术下智能合约之于合同的新诠释[J]. 重庆大学学报(社会科学版),2021,27(5):169-182. DOI:10.11835/j.issn.1008-5831.fx.2020.04.001.

致　谢

时光飞逝，转眼间我的四年本科生涯即将结束，在进入研究生学习之前，回望过去，我的求学生涯得到了非常多人的帮助。从初中开始我便得到了来自国家的贫困补贴，上了大学还进一步申请了助学贷款，可以说没有国家的资助，我的求学之路远没有现今那么轻松。在大学四年的学习中，我十分感谢各位辅导员和老师的帮助和教导，是他们带领我不断拓宽视野，汲取更多的知识，解决在学习中遇到的种种困难。同时，我更要感谢许光全老师对我的帮助，从大二开始，我就跟随许老师完成大学生创新创业项目，还在老师的领导下完成实验室的课题，许老师给予我的不仅是学习和锻炼的机会，还带领我走上了研究生的道路。在此衷心感谢许老师的悉心教导。

除此之外，我的毕业设计的完成离不开雷文清师兄和老师、同学、家人的帮助，感谢雷文清师兄在课题研究时提供的指导和帮助，也感谢我的同学帮忙解决实验中遇到的问题。

最后，再次衷心感谢以上提到的所有人。