软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(10):3546–3564 [doi: 10.13328/j.cnki.jos.006627]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

FactChain: 一个基于区块链的众包知识融合系统*

朱向荣, 吴鸿祜, 胡 伟

(计算机软件新技术国家重点实验室(南京大学), 江苏 南京 210023)

通信作者: 胡伟, E-mail: whu@nju.edu.cn

摘 要: 知识图谱作为诸多人工智能应用的关键, 受到学术界和工业界的广泛关注. 当前的知识图谱一般由特定组织构建并维护, 以RDF转储文件或SPARQL查询接口的方式提供知识访问服务, 这种中心化的管理方式存在不能持久化访问的弊端. 具体来说, 一旦服务提供者单点崩溃, 用户就无法以可靠的方式获取知识. 此外, 知识因时效性可能需要更新, 不同来源的知识之间可能存在冲突, 传统的知识图谱构建维护方式难以有效地处理这些问题. 区块链技术以其分布式存储与共识机制, 为知识图谱的分布式构建与管理提供了新思路. FactChain是一个基于区块链的知识管理系统, 具有为知识的多源共享与融合建立全新的去中心化生态的潜力. 使用联盟链作为底层架构, 由区块链、组织和参与人这3层结构组成. 通过区块链上的智能合约编程实现融合多源冲突知识的真值验证算法, 具有在组织层面实现并部署基于分布式应用的参与人管理、在本地局部本体与全局共享本体间建立映射以及结合链上与链下数据响应参与人查询请求等功能.

关键词: 知识图谱; 区块链; 分布式知识管理; 知识融合
中图法分类号: TP18

中文引用格式: 朱向荣, 吴鸿祜, 胡伟. FactChain: 一个基于区块链的众包知识融合系统. 软件学报, 2022, 33(10): 3546–3564. http://www.jos.org.cn/1000-9825/6627.htm

英文引用格式: Zhu XR, Wu HH, Hu W. FactChain: A Blockchain-based Crowdsourcing Knowledge Fusion System. Ruan Jian Xue Bao/Journal of Software, 2022, 33(10): 3546–3564 (in Chinese). http://www.jos.org.cn/1000-9825/6627.htm

FactChain: A Blockchain-based Crowdsourcing Knowledge Fusion System

ZHU Xiang-Rong, WU Hong-Hu, HU Wei

(State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023, China)

Abstract: Knowledge graphs (KGs) have drawn massive attention from both academia and industry, and become the backbones of many AI applications. Current KGs are often constructed and maintained by large parties, which provide services in the form of RDF dumps or SPARQL endpoints. This kind of centralized management has inherent drawbacks like non-durable accessibility. Furthermore, some facts in KGs may be outdated or conflicting, and there is no convenient way of resolving them democratically. As an innovative distributed infrastructure, blockchain has many characteristics such as decentralization and consensus, which is of great significance for the construction and management of KGs. This study designs a blockchain-enhanced knowledge management framework called FactChain, which aims to establish a new decentralized ecology for knowledge sharing and fusion. FactChain leverages a consortium architecture containing blockchain, organizations, and participants. The on-chain smart contracts enable the truth discovery algorithm of multiple-source conflicting knowledge. FactChain also supports participant management, mapping between local schemata and global ontology and integration of on/off-chain knowledge based on the decentralized application (DApp) in organizations.

Key words: knowledge graph; blockchain; distributed knowledge management; knowledge fusion

* 基金项目: 国家自然科学基金(61872172)
本文由“智慧信息系统新技术”专题特约编辑邢春晓研究员、王鑫教授、张勇副研究员、于戈教授推荐.
收稿时间: 2021-07-19; 修改时间: 2021-08-30, 2021-12-24; 采用时间: 2022-01-14; jos在线出版时间: 2022-02-22

© 中国科学院软件研究所 http://www.jos.org.cn

1 引 言

自 Tim Berners-Lee 提出语义网[1]的概念, 到谷歌发布 Google Knowledge Graph 至今, 知识图谱已经成为语义搜索、智能问答和推荐系统等诸多人工智能应用的关键支撑. 知识图谱以结构化的方式组织、描述和理解客观世界中的概念、实体及其之间的关系, 是结构化的语义知识库. DBpedia[2]、Wikidata[3]以及 Probase[4]等知名的大型知识图谱包含百万级别的实体和十亿级别的知识, 通常由特定组织经过数据收集与清洗、知识抽取与融合等流程构建并以版本更迭的形式进行中心化管理, 一般以 RDF (resource description framework)转储文件或 SPARQL 查询接口的形式提供知识访问服务.

然而, 这种中心化的构建与管理方式存在固有的弊端. 譬如, 如果一个知识图谱的服务站点崩溃, 则该知识图谱无法被访问, 即中心化的存储难以应对单点崩溃故障. 又如, 知识图谱中的一些知识可能随时间发生变化, 并且知识之间还可能存在冲突, 传统知识图谱通过发布新版本来应对上述问题, 鲜有便捷的增量式解决方案. 比如, 截至本文写作时, 最新版本的 DBpedia 仍将拜登(Joe Biden)作为美国副总统. 另外, 随着知识图谱的普及, 用户生成内容和传感器数据这些频繁更新的数据也被组织成知识图谱的形式, 传统的中心化构建与管理方式难以适应这种持续更新的数据特征.

作为以比特币(bitcoin)[5]为代表的电子货币的底层技术, 区块链(blockchain)[6]是一种创新性的分布式架构和计算范式. 区块链具有去中心化、开放、透明、可追溯、不可篡改等特性, 被广泛应用于证券交易[7]、电子商务[8]、物联网[9]以及其他许多领域. 本文认为, 上述区块链的特性亦能为知识图谱的构建与管理创造一种全新的生态, 有利于解决上面提到的知识图谱的中心化构建与管理方式的弊端. 具体地, 本文提出一个基于区块链的众包知识融合系统 FactChain, 其关键技术与主要贡献如下.

(1) FactChain 具有基于联盟链的 3 层架构, 分别为区块链、组织和组织内的参与人. 区块链的去中心化和开放性保证了系统的单点崩溃容错性质; 区块链的可溯源和不可篡改性使得知识的贡献者和更新流程可追溯. 此外, 基于联盟链的 3 层架构保证了 FactChain 作为一个分布式系统达成共识的效率.

(2) FactChain 通过编程区块链上的智能合约(smart contract)实现链上的知识融合逻辑, 保证了链上知识的一致性访问. 具体地, 提出了一种置信度加权投票算法和一种垄断分红算法以自动化地执行众包知识融合中的真值验证和激励过程.

(3) 针对多值知识和随时间变更的知识这两种真值验证现实问题, 在系统中设计专门的策略加以解决. 具体地: 提出了先推断真值数量再确定真值集合的方法以解决多值知识的真值验证问题. 在链上采用统一的时序知识表示方法, 设计了结合前一版本真值确定下一版本真值的策略以处理随时间变更的知识融合问题.

(4) 为了提高系统的存储能力和保障私有数据隐私, FactChain 区分全局链上知识和组织级别链下知识. 在组织层面实现分布式应用程序(decentralized application, DApp), 具有参与人管理、链上链下模式转换以及查询应答功能, 从而支持结合链上和链下知识的查询和推理.

本文第 2 节介绍区块链的相关概念以及基于区块链的相关知识管理系统. 第 3 节展示 FactChain 的系统架构, 并解释架构的设计思路. 第 4 节描述 FactChain 的设计细节, 包括系统的交互流程以及底层真值验证算法等. 第 5 节报告在真实数据集上进行的 FactChain 性能和效果测试结果. 第 6 节总结全文并讨论未来工作.

2 相关工作

本节首先介绍区块链的基础概念, 然后概览基于区块链的相关知识管理系统.

2.1 区块链

区块链使用块链式数据结构存储和验证数据. 每一个区块由区块头和区块体组成. 区块头存储了当前区块的元数据和上一个区块的哈希值, 而随时间不断增长的交易数据被组织成区块体. 通过每个区块包含上一区块哈希值的密码学方式将不断生成的交易数据组织成不可篡改的链. 区块链中的共识机制[10]旨在保证数据的最终一致性. 共识机制的研究最初源自分布式系统中对容错机制的探索, 其本质是分布式节点之间建立信

© 中国科学院软件研究所 http://www.jos.org.cn

任、分发权益并达成一致的算法. 区块链系统一般使用密码学方法保障数据传输和访问的安全. 比特币系统使用椭圆曲线电子签名算法进行比特币的确权, 使用SHA-256哈希函数和Merkle树结构保护交易数据不可篡改. 区块链系统提供给外部的编程和操纵数据调用接口被称作智能合约. 智能合约[11]是预先定义好的确定性业务逻辑, 经过验证后被部署在区块链系统中的所有分布式节点上, 在用户调用后自动且安全地执行以达成分布式一致性.

根据开放程度的不同, 区块链系统被分为公有链(public blockchain)、联盟链(consortium)和私有链(private blockchain). 公有链又称为无许可链(permissionless blockchain), 公有链的概念设计愿景是完全去中心化、不受任何区块链参与者控制, 网络上任何人均可参与. 比特币系统和去中心化应用平台以太坊(Ethereum)[12]都是公有链的代表. 联盟链又被称作许可链(permissioned blockchain), 仅供注册过有证书的联盟成员加入. 联盟链的规模灵活, 联盟成员可以是非政府组织或者国家地区等. 知名的联盟链包括多家国际银行组织共同建立的 R3 Alliance 和由 Linux 基金会维护的开源项目 Hyperledger[13]等. 私有链的参与者被限定在一个特定的国家、企业、组织乃至个人的范围内, 一般目的在于提供一个安全可追溯、不可篡改的存证和智能合约执行平台.

上述3种区块链分别适合不同的应用场景, 各有利弊. 比特币作为首个点对点的电子现金区块链, 开放给全网用户参与. 参与者必须付出巨大的算力代价以获取创建新区块的记账权和记账所得的交易费. 比特币系统中这种凭借算力竞争记账权的共识机制被称为工作量证明(proof-of-work, PoW). 工作量证明机制虽然确保了比特币系统中数据的安全性和一致性, 但却牺牲了记账效率. 由于用户准入机制, 联盟链和私有链可以选用更加宽松的共识机制, 从而获得更高的效率, 更加适合追求高吞吐率和低延迟的应用场景.

2.2 区块链与知识管理

现有的结合区块链和知识图谱的工作主要有两种路线.

(1) 使用知识图谱来增强区块链上数据的语义表达能力[14−16];

(2) 在区块链的架构之上实现知识管理系统[17−22].

FactChain 属于第 2 种路线. 相比于传统的知识管理系统, 区块链因采用去中心化的架构, 可以从底层保证电子资产在创建和传递过程中的不可篡改. 知识图谱中的知识可以被收集、整合、管理和共享, 正如区块链上的电子资产. 因此, 知识图谱和区块链可以和谐共生、相互促进.

表 1 从 4 个方面对比了近期一些使用区块链实现知识管理系统的工作, 包括底层区块链的类型、知识管理的内容、是否整合链上和链下的数据, 以及具体的应用场景. 下面简要介绍相关工作.

• Knowledge Blockchains[17]是一个面向企业建模的知识管理系统. 由于数据隐私的要求, 该系统采用私有链. 使用区块链存储业务流程模型和标记法(business process model and notation, BPMN)文件, 便于透明地监控文件的变化、追溯源头和验证其存在性.

• GraphChain[18]是一个法律实体标识符(legal entity identifier, LEI)管理系统的区块链平台. GraphChain 的底层数据基于 RDF 图, 使用一种 RDF 序列化方法生成每个 RDF 图的哈希值作为其摘要. GraphChain 支持以标准 Web 协议(例如 Http)的方式发布和访问数据.

• Knowledge Market[19]是一个知识有偿共享的区块链平台, 旨在打破数据孤岛, 并交易网络边缘的人工智能物联网设备上的知识. Knowledge Market 使用联盟链架构以保障数据的安全性和系统的效率. 物联网传感器上传的数据流被网络边缘人工智能设备训练成为机器学习模型. 这些机器学习模型被视作知识商品在区块链上存储交易.

• AUDABLOK[20]使用公有链整合 CSV、JSON、XML 等格式的开放政府数据和人工算力, 以促进市民自觉参与政府的开放数据接口构建. AUDABLOK 为综合知识档案网络(comprehensive knowledge archive network, CKAN)提供基于区块链的增强功能, 支持对其中的数据集的变更请求进行全面审查和管理.

• TRUSTD[21]是一个基于公有链的社交媒体平台, 将发布和审核内容的权限分发给平台用户, 使用集体签名技术来追溯内容权责, 旨在创建一个用户主动且自觉打击虚假内容的社交媒体生态. TRUSTD 允许平台

© 中国科学院软件研究所 http://www.jos.org.cn

用户指定自己信任的内容发布者、标注特定发布者和内容的可信度级别, 以帮助平台用户判别内容可信度, 助力内容创建者获得社区支持.

• OpenKG Chain[22]旨在基于区块链建设开放知识共享的基础设施. OpenKG Chain 由共享知识图谱数据集和工具集等粗粒度资源的 OpenKG.CN、众包开放知识图谱 CnSchema 和细粒度三元组众包平台 OpenBase 组成, 定义了 *K*-Point 和 OpenKG Token 用以分别度量知识的价值和参与者的贡献.

表 1 基于区块链的知识管理系统相关工作

	Blockchain types	Knowledge management content	Integrating on/off-chain data	Applications
Knowledge Blockchains	Private	BPMN file	×	Enterprise modeling
GraphChain	Private	RDF graph	×	Legal entity identifier system
Knowledge Market	Consortium	Macheine learning model	×	IoT knowledge trading
AUDABLOK	Public	Open format (CSV, JSON, XML)	×	Open government data refinement
TRUSTD	Public	Multimedia	×	Fake content combat
OpenKG Chain	Consortium	Data/tool set & triple	√	Knowledge graphs sharing
FactChain (ours)	Consortium	RDF triple	√	Knowledge fusion

上述相关工作主要重心在于基于区块链建立知识管理系统, 仅 OpenKG Chain 和 FactChain 考虑了链上和链下数据的整合. 由于区块链上的数据透明共享且有冗余地复制到各个分布式节点, 部分不具备共享价值或者有保密需求的数据应当被存储在链下. 因此, 结合链上和链下数据的访问和管理功能是系统的一个基本需求. FactChain 设计基于联盟链的 3 层架构, 在区块链上通过智能合约编程解决知识融合中的真值验证[23]现实问题, 达成跨组织的知识融合的目标. 在组织层面以链下存储模块维护本地知识图谱, 扩充系统的存储能力, 并提供结合链上与链下知识的查询与推理功能. OpenKG Chain 和 FactChain 的主要区别在于 OpenKG Chain 以数据集级别的粗粒度资源共享为主, 侧重于知识的确权和价值度量, 设计了 *K*-Point 和 OpenKG Token 的度量指标. 而 FactChain 侧重于细粒度知识的融合与推理, 设计了一系列方法和技术以解决多源冲突知识的真值验证问题.

3 系统架构

FactChain 作为一个基于区块链的众包知识融合系统, 其中, 区块链是系统的架构基础, 众包是系统的数据获取模式, 知识融合是系统承载的功能. FactChain 在实现知识融合功能时的众包激励特性和去中心化、可溯源等性质是由图 1 展示的参与人、组织和区块链这 3 层架构及层与层之间的交互共同实现的. 下面概述 FactChain 的 3 层架构中的角色定义和主要功能.

角色. FactChain 为 3 层架构, 存在参与人、组织和区块链 3 种角色.

• 参与人作为组织内部的成员, 是系统中的众包工作者、知识的贡献者和交互的发起者, 可以共享知识以获取收益或者付出报酬请求知识. 对于参与人, 组织内部的链下知识、链上共享的知识和自身的信用额度是可见的, 知识融合的具体过程是不可见的.

• 组织是连接区块链和参与人的桥梁, 其中包含分布式应用、区块链对等节点(peer node)和链下存储等模块. 组织具有封装在 X.509 数字证书中的数字身份, 保有合法身份证书的组织通过对等节点加入区块链, 由区块链确定其对资源和信息的访问权限. 组织通过分布式应用管理参与人的身份及其与 FactChain 系统交互的权限, 响应参与人的交互请求.

• 区块链是 FactChain 架构的基础, 由链上存储模块和智能合约组成. 链上存储模块负责储存和管理组织间共享的知识, 以密码学方法保障知识的可溯源和不被篡改. FactChain 通过智能合约编程实现预定义的知识融合应用中的真值验证和激励分配算法, 提供给组织根据参与人请求调用智能合约的接口.

© 中国科学院软件研究所 http://www.jos.org.cn

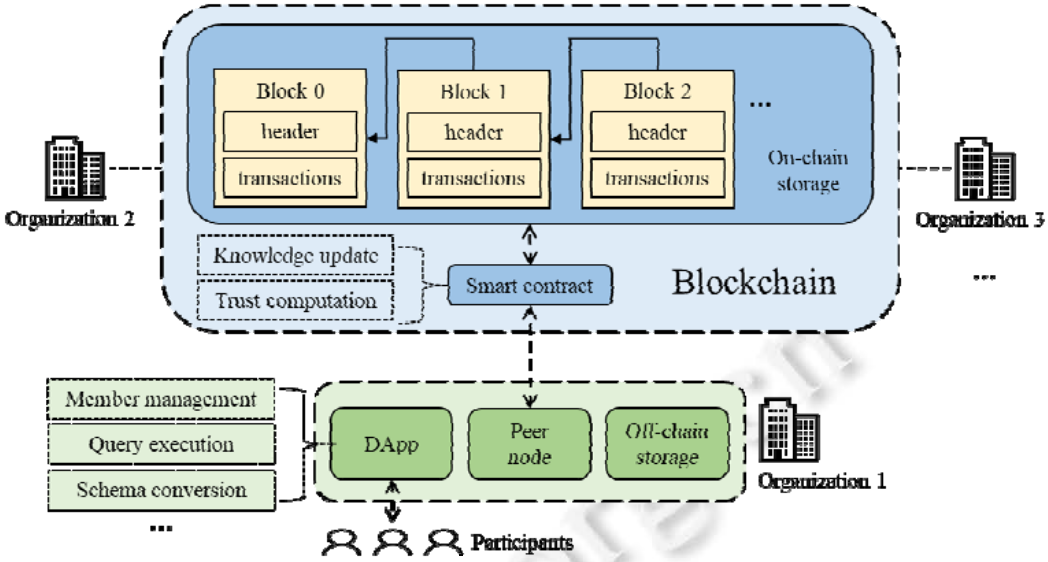


图 1 FactChain 架构图

功能. FactChain 的功能可以分为日常管理和知识服务两类.

- 日常管理功能包括区块链对组织的管理和组织对组织内参与者的管理. 区块链生成、发放和审核对等节点身份证书, 管理组织参与区块链网络的权限. 组织通过 Authorize 命令为组织内参与人在 FactChain 中登记身份, 通过 Deauthorize 命令注销参与人身份.
- 知识服务功能是 FactChain 的现实价值, 参与人向所属组织发起请求, 经过组织与区块链交互, 达成众包知识图谱的构建共享与多源冲突知识融合的目标.

(1) 知识查询. 参与人可以查询链上共享的知识和所属组织的链下知识图谱. FactChain 通过组织层面的分布式应用实现了结合链上与链下知识的查询与推理功能.

(2) 知识众包. 对于不存在于链上与链下的知识, 参与人可以通过 FactChain 向不同组织的参与人发起众包请求, 付出一定的报酬, 激励其他参与人助力众包知识图谱的构建.

(3) 真值验证. 参与人可以浏览系统中正被众包请求的知识, 根据私有知识为自己支持的候选结果投票. 不同参与人对同一个众包请求可能贡献不同的候选结果. FactChain 实现了置信度加权投票算法和垄断分红算法, 以解决多源冲突知识的融合问题.

FactChain 架构的设计源自对去中心化知识管理的思考与理解. 公有链如比特币的理论模型是完全去中心化的. 然而, 由于存储能力的限制, 只有存储全量数据的全节点拥有记账权, 因而成为了比特币网络上的局部中心. 轻节点仅存储区块头, 只具有在区块头上进行简单支付验证(simple payment verification, SPV)的能力, 没有挖掘新区块的记账权. 由于工作量证明机制需要付出庞大的算力来竞争记账权, 现实世界中的比特币网络成为了一个多中心网络, 拥有庞大算力的组织成为比特币网络中的区域中心. 不同于概念上完全去中心化的公有链, FactChain 基于联盟链设计. 区块链负责组织间透明的知识管理. 组织作为区块链网络的实际参与者, 扩展了区块链网络的存储能力, 同时负责对组织内参与人的管理和响应. 在这种 3 层架构之下, 组织之间形成了一个去中心化的存储和计算的区块链网络, 各组织和组织内参与人建立了一个交互上的多中心系统.

4 设计细节

本节描述 FactChain 的众包交互逻辑以及示意流程, 然后介绍链上基于智能合约的知识融合方法及面向真值验证现实问题的扩展, 最后简述链下分布式应用程序的功能. 表 2 列举了本文常用的符号及其含义.

© 中国科学院软件研究所 http://www.jos.org.cn

表 2 符号定义

符号	描述	符号	描述
org	组织	(s,p,o)	描述知识的三元组, 其中, s 称为主语, p 称为谓语, o 称为宾语
$ptcp$	参与人, 有唯一组织		
b	参与人信用额度	r	发起众包请求的激励
$acct=(org,ptcp,b)$	参与人账户	c	投票附加的置信度
–	–	$t=(s,p,?,r)$	对三元组的众包请求
–	–	$t^*=(s,p,o^*)$	众包请求$(s,p,?,r)$经过验证的真值
–	–	$v=(org,ptcp,s,p,o,c)$	$ptcp$ 对$(s,p,?,r)$的投票

4.1 交互逻辑

从数据管理的视角看, 区块链是一种以数据冗余来换取崩溃容错性的分布式数据库. 区块链中的电子资产即数据库上的状态, 区块链的账本即对数据库状态操纵修改的日志. 相应地, 智能合约对应于对数据库进行简单读写之外的复杂操作接口, 而共识机制是保证分布式数据库节点上数据最终一致性的算法.

图 2 给出了 FactChain 从数据管理视角定义的交互逻辑的概要. 其中, 图 2(a)概览了几种主要的全局状态结构, FactChain 系统中的分布式节点在这些状态上保持一致性. 分别为记录组织内成员账户信息的 Account 结构、记录众包请求的三元组元数据的 Triple 结构、参与人根据链下知识响应众包请求提交的候选结果 Vote 结构和在众包任务上经过真值验证流程最终达成一致的 Answer 结构.

图 2(b)概述了提供给组织调用以查询操纵链上状态的智能合约接口. 其中, Authorize 和 Deauthorize 是组织对参与人在区块链上权限的管理操作. 组织使用 Authorize 命令为参与人在系统中注册账户, 参与人方可参与系统交互. 组织使用 Deauthorize 命令注销参与人的账户之后, 参与人便无权在系统中发起众包请求或者对其余参与人的众包请求为自己支持的候选结果投票. Request、Commit 以及 Query 操作由参与人发起, 经过组织预处理之后提交到区块链层面执行智能合约, 合约执行的结果经由组织传回参与人. 下面结合图 2 和图 3 描述经过简化的 FactChain 运行流程.

(1) Authorize. 在参与人与 FactChain 系统交互之前, 组织需要使用 Authorize 命令为参与人在系统中注册账户, 分配固定的账户余额. FactChain 在链上会添加相应参与人账户的 Account 状态.

(2) Request. 当参与人需要的知识无法在系统中(链上/链下)查询获得时, 参与人通过 *Request*(*s*,*p*,?,*r*)命令发起众包请求. 组织收到请求后在其上添加表征组织和参与人身份的信息, 将其扩充成为 *Request*(*org*,*ptcp*,*s*,*p*,?,*r*), 调用相应的智能合约. FactChain 在链上会扣除相应参与人账户余额, 生成对应众包请求的 Triple 状态.

(3) Commit. 当参与人拥有正在链上被众包请求的知识时, 可以通过 *Commit*(*s*,*p*,*o*,*c*)命令为自己支持的候选结果投票以竞争收益, 代表其以确信程度 c 支持候选结果 o. 组织对命令添加组织与参与人的信息, 将其扩充成为 *Commit*(*org*,*ptcp*,*s*,*p*,*o*,*c*), 调用智能合约. 在链上会扣除相应参与人账户余额, 生成对应投票提交的 Vote 状态.

(4) Query. 参与人可以通过 *Query*(*s*,*p*,?)命令查询链上和链下知识. 图 2(b)展示了通过智能合约实现的链上知识查询过程: 首先查找相应 Triple 对应的 Answer 是否存在. 若存在, 即返回 Answer 状态. 若不存在, 且对应的 Vote 满足要求, 则进入真值验证流程, 生成相应的 Answer 状态并返回.

FactChain 系统上的查询不仅限于图 2(b)所示的链上逻辑, 完整逻辑由组织层面的去中心化应用负责, 如图 3 的步骤 4 所示. 组织一方面在本地存储的链下知识图谱执行查询, 一方面调用智能合约查询链上知识, 最后整合链上和链下知识返回给参与人.

(5) Deauthorize. 当组织认为参与人不再适合参与 FactChain 系统的交互时, 组织可以通过 Deauthorize 命令调用智能合约, 注销参与人在系统中的账户. FactChain 在链上删除相应的 Account 状态.

© 中国科学院软件研究所 http://www.jos.org.cn

State	
Account	
Org	organization the member belongs to
Member	identifier of the member
Balance	account balance of the member
Triple	
Subject	subject of the triple
Predicate	predicate of the triple
TimeCounts	true if time counts
MultiValue	true if object is multi-valued
IsSolid	true if answer is solid
Reward	reward the requestor paid
Requestor	member requested the triple
Answer	
Subject	subject of the triple
Predicate	predicate of the triple
Object	object of the triple
TimeStamp	null if TimeCounts==false
Supporters[]	supporters of the object
Vote	
Subject	subject of the triple
Predicate	predicate of the triple
Object	candidate object of the triple
Voter	voter of the candidate object
Confidence	confidence of the voter to the vote
Checked	true if the vote is checked
TimeStamp	null if TimeCounts==false

(a) 状态概要

Interface	
Authorize	if already exist, return error; otherwise, put account with balance into state.
Deauthorize	if not exist, return error; otherwise, delete account from state.
Request	
Arguments	requestor, reward, timecounts, multivalue, subject, predicate
Results	error
Logic	if triple exist, return error; if requestor not exist, return error; if balance<reward, return error; otherwise, decrease balance, put triple into state.
Commit	
Arguments	voter, confidence, timestamp, subject, predicate, candidate
Results	error
Logic	if triple not exist, return error; if triple is solid, return error; if balance<confidence, return error; otherwise, decrease balance, put vote into state.
Query	
Arguments	subject, predicate
Results	answer, error
Logic	if triple not exist, return error; if triple solid, return answer; if not enough votes, return error; otherwise, perform truth-discovery, assign reward to voters, change triple state, put answer into state, return answer.

(b) 接口概要

图 2 FactChain 交互逻辑概要

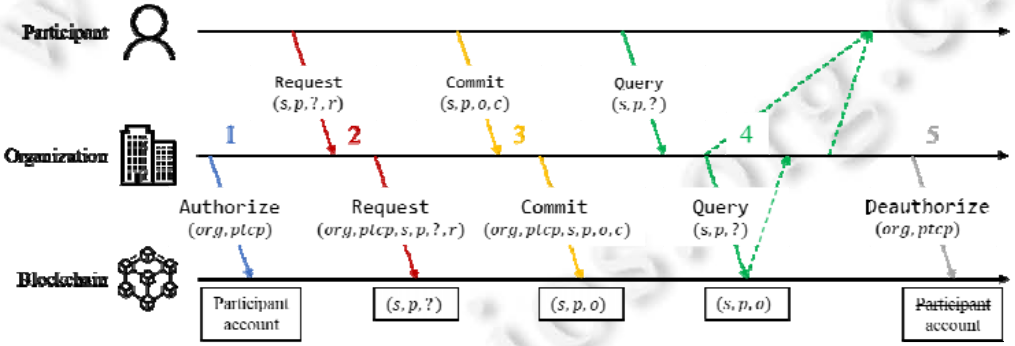


图 3 FactChain 的概念流程

FactChain 中账户额度的付出与获取途径设计源自对众包系统和区块链系统中激励的理解与思考. 在众包系统的设计中, 发起众包请求需要付出代价, 完成众包任务会获得激励. 区块链系统中需要对守序成员加以激励, 对恶意成员加以惩罚. 因此在 FactChain 中, 发起众包请求需要一定的账户余额作为众包赏金, 对众包任务提交候选结果需要付出一定额度的账户余额作为担保, 这个额度既充当真值验证算法中参与人对自己支持的候选结果的确信程度, 同时影响激励分配时参与人分配到的奖励额度. 查询数据不需要付出代价, 因为知识的贡献者在真值验证流程完成之后的激励分配过程中已经获得了分红, 而对共享知识去中心化的无限制访问符合知识图谱本身自由开放的技术精神.

© 中国科学院软件研究所 http://www.jos.org.cn

4.2 基于智能合约的知识融合

由于不同组织的本地链下知识图谱不同, 以及不同的参与人的私有知识不同, 不同参与人对同一个众包请求给出的候选结果可能存在冲突, FactChain 聚焦于解决冲突知识的融合. 多源冲突知识的融合问题又被称作真值验证问题. 基于表 2 符号定义, 本文对知识图谱上的真值验证问题描述如下: 来自不同组织的参与人集合 $P=\{ptcp_1,ptcp_2,\ldots,ptcp_{N_p}\}$ 对三元组众包请求集合 $T=\{t_1,t_2,\ldots,t_{N_t}\}$ 给出了可能包含冲突的投票集合 $V=\{v_1,v_2,\ldots,v_{N_v}\}$. 真值验证的目标是消解 V 中的冲突, 为众包请求集合 T 生成三元组真值集合 $T^*=\{t_1^*,t_2^*,\ldots,t_{N_t}^*\}$, 并将众包请求激励与投票置信度的总和 $\left(\sum_{(s,p,?,r)\in T}r+\sum_{(org,ptcp,s,p,o,c)\in V}c\right)$ 合理分配给 P 中参与人.

现有的真值验证问题解决方案大多遵循以下假设: 频繁提供真实信息的数据源具有更高的置信度, 而被更多高置信度数据源支持的信息更可能接近真实信息. 基于上述基本假设, 真值验证算法在没有监督数据的情况下估算出数据源的置信度和存在冲突的问题的真值. 根据对基本假设的建模方式的不同, 常见的真值验证算法大致分为 3 类[23]: 基于迭代的方法、基于优化的方法和基于概率图模型的方法.

FactChain 通过处理参与人的众包、投票和查询请求, 迭代(1)执行置信度加权投票算法计算众包问题真值和迭代(2)根据垄断分红算法将信用额度回报给参与人这两个步骤来实现迭代式的真值验证算法, 以解决知识融合的真值验证问题. FactChain 中每个组织内的参与人注册时被赋予一定的信用额度, 信用额度兼具真值验证问题中数据源置信度的作用和众包模式中的激励意义. 在步骤(1)中结合参与人提交投票时对候选结果附加的信用额度计算目标问题的预测真值, 在步骤(2)中将目标问题的众包报酬和所有投票上附加的额度作为奖励合理分发给知识贡献者.

采取基于迭代的方法而非更加复杂的基于优化或者概率图模型方法, 是由 FactChain 的数据特征和系统架构的一致性要求决定的. FactChain 中的参与人概念对应现实生活中参与跨组织知识图谱构建的组织内私有知识的拥有者, FactChain 中共享的数据由参与人的私有知识和链下存储的组织内私有知识组成, 组织内私有知识一般是由部署在组织内的知识抽取程序提供. 相比于传统真值验证问题, FactChain 系统的数据来源具有更大程度的人工贡献和较少的随机噪声, 更加结构化和规范化, 使用迭代式算法能够很好地利用群体智慧解决真值验证问题. 另一方面, FactChain 系统采用联盟链架构, 区块链上的智能合约是一种自动执行预定义逻辑的程序脚本, 为了保证分布式节点上执行相同的运算并达成共识, 智能合约必须采用确定性逻辑. 基于优化或概率图模型的真值验证算法都是机器学习算法, 其输出是以训练数据为经验在某些概率分布下产生的, 具有随机性, 可能造成结果的不一致.

4.2.1 置信度加权投票算法

FactChain 中的置信度加权投票算法详见算法 1, 下面详述计算过程. 给定一个对三元组的众包请求 t=(s, p,?,r), 将其对应的 n 个候选结果表示为集合 $O=\{o_1,o_2,\ldots,o_n\}$. 对于每个候选结果 $o_i\in O$, 共有 m 个参与人为其投票, 将相应置信度表示为集合 $C_{t,i}=\{c_{i,1},c_{i,2},\ldots,c_{i,m}\}$, 其中, $c_{i,j}$ 表示第 i 个候选结果的第 j 个投票者对该次投票付出的置信度. 根据以下公式计算每个候选结果的置信度得分, 根据算法 1 选择得分最高的候选结果作为该众包任务的最终结果 o^*.

算法 1. 置信度加权投票算法.

输入: $t=(s,p,?,r)$, V_t;

输出: $t^*=(s,p,o^*)$.

1. $O=\{o\mid(org,ptcp,s,p,o,c)\in V_t\}$;
2. **for** i=1 **to** $|O|$ **do**
3. 根据公式(1)计算 $\hat{c}_i$, $\sigma(\hat{c}_i)$;
4. 根据公式(2)计算 $score(o_i)$;
5. **end**

© 中国科学院软件研究所 http://www.jos.org.cn

6. $o^* = \text{argmax}_{o_i \in O}(score(o_i))$;

7. **return** $t^* = (s, p, o^*)$;

$$\hat{c}_i = \frac{1}{m}\sum_{j=1}^{m} c_{i,j},\ \sigma(\hat{c}_i) = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(\hat{c}_i - c_{i,j})^2} \tag{1}$$

$$score(o_i) = \frac{\sum_{j=1}^{m} c_{i,j}}{1+\sigma(\hat{c}_i)} \tag{2}$$

公式(1)计算置信度的均值和标准差. 公式(2)以标准差为偏置项, 计算候选结果的置信度得分. 采用这种以标准差为偏置的置信度加权投票算法主要是为了解决多个获得相同额度投票的候选结果之间的最佳结果选择问题. 在多个候选结果获得的总投票分数相同的情况下, 以标准差为偏置项的变种倾向于选择从不同的参与人处获得的置信度投票相近的候选结果. 即认为支持者意见更为一致的候选结果更有可能是更好的结果.

4.2.2 垄断分红算法

交易费对比特币交易发挥着愈发重要的作用, 在比特币定价机制研究中, 垄断定价机制(monopolistic price, MP)[24]经实验证实优于随机抽样最优定价机制(random sampling optimal price, MSOP)[25]. 可证明[26], 对于任何独立同分布情况, 随着区块链网络中用户数量的增大, 垄断定价机制具有近似激励相容(nearly incentive compatible, nearly-IC)的性质, 即个人利益与集体利益趋于一致.

FactChain 中的账户额度兼具真值验证中加权投票的置信度、众包模式中对知识共享的激励[27]和区块链中的交易费3种属性, 目的在于奖励守序积极的知识贡献者, 抑制恶意自私的参与人, 促进系统在无人干预监管的情况下健康运行. 受垄断定价机制启发, FactChain 设计了垄断分红算法, 具有近似激励相容性质, 同时能免于拆分投标(splitting bids)的恶意攻击.

FactChain 的垄断分红算法详见算法 2, 下面给出补充说明. 算法 2 第 1 行重排后 $V_t^* = \{(ptcp_1, c_1), (ptcp_2, c_2), \ldots, (ptcp_m, c_m)\}$, $c_1 \geqslant c_2 \geqslant \ldots \geqslant c_m$. 公式(3)计算垄断分红的参与人编号阈值 k^*. 公式(4)计算参与人获得的分红, 序号小于等于阈值 k^*的参与人获得自身初始投入的置信度和额外分红, 序号大于阈值 k^*的参与人仅收回自己初始投入的置信度而不获得额外分红.

为了更加简明地阐释垄断分红算法, 表 2 展示了一个应用垄断分红算法来分配激励的例子. 参与人及其付出的置信度按照置信度 c_k降序重新排列. 当 k=3 时, $k \times c_k$取到最大值 1.8. 因此, 垄断分红的阈值设置为 k^*=3, 序号小于等于 3 的参与人获得多于投入置信度的激励, 而最终结果的其余支持者仅收回初始投入的置信度. 假设其余候选结果获得的置信度以及该众包任务的发起者付出的报酬的总和为 4.8, 根据垄断分红算法分配的方案见表 2.

算法 2. 垄断分红算法.

输入: C, V_t^*, A;

输出: A.

1. 按置信度降序重排 V_t^*;
2. $P_t^* = \{ptcp | (ptcp, c) \in V_t^*\}$;
3. 根据公式(3)计算 k^*;
4. **for** j=1 **to** $|P_t^*|$ **do**
5. $\quad acct_j = (org_j, ptcp_j, b_j) \in A$;
6. $\quad$ 根据公式(4)计算 $dividend(ptcp_j)$;
7. $\quad b_j += dividend(ptcp_j)$;
8. **end**
9. **return** A;

$$k^* = \underset{k \in [1,m]}{\text{argmax}}(k \times c_k) \tag{3}$$

© 中国科学院软件研究所 http://www.jos.org.cn

$$dividend(ptcp_j)=\begin{cases}\dfrac{C-\sum_{k=1}^{m}c_k}{\sum_{k=1}^{k^*}c_k}\times c_j+c_j, & 1\leqslant j\leqslant k^*\\ c_j, & k^*<j\leqslant m\end{cases} \tag{4}$$

按照垄断分红算法, 若参与人投入的置信度过少, 即使其支持了正确的候选结果, 最终仍然可能只收回最初付出的投入而不获得额外的分红. 这种机制更有利于付出更高置信度的参与人, 不利于付出较低置信度的参与人. 考虑到该众包系统的实际参与人是会权衡利弊的个体, 这种方案能够推动参与人在提交候选结果时更加谨慎, 为自己相信的可靠候选结果付出较大的置信度, 尽量不为不确信的候选结果投票, 从而促进守序的知识共享. Lavi 等人[24]证明, 在 FactChain 的垄断分红算法下, 当参与人数量有限且尽可能大时, 参与人无法通过恶意行为获得比守序行为更高的收益.

表 2 垄断分红算法的示例

k	1	2	3	4	5	…
c_k	1.0	0.8	0.6	0.3	0.2	…
$k\times c_k$	1.0	1.6	1.8	1.2	1.0	…
divident	3.0	2.4	1.8	0.3	0.2	…

4.2.3 真值验证算法的区块链实现

传统的迭代式真值验证算法一般在全量数据上执行多次迭代直至数据源置信度和目标问题预测值收敛, 输出此时的数据源置信度和目标问题预测值作为结果. FactChain 中的多源冲突知识是由用户与系统的交互不断产生的增量式数据, 针对这种增量式数据特征, FactChain 将传统迭代式真值验证算法拆分为第 4.2.1 节中的置信度加权投票算法和第 4.2.2 节中的垄断分红算法的迭代执行, 详见算法 3.

第 4.1 节介绍了 FactChain 提供给参与人与系统交互的 Request、Commit 和 Query 命令, 图 2 简要叙述了这些命令的输入输出和响应逻辑以及在区块链中生成与更新的相应状态的数据结构. 具体来说, 在目标问题的真值验证流程中, 首先获取该目标问题对应的所有未被验证的投票数据 Vote (checked=False), 通过置信度加权投票算法计算目标问题的预测值, 同时得到为目标问题最终预测值投票的参与人列表, 根据垄断分红算法将激励分配给这些参与人, 更新相应的 Account. 在上述两个流程中, 只需要从链上存储获取目标问题对应的未被验证的投票 Vote 和为最终预测值投票的参与人的 Account, 避免了链上数据的冗余访问, 减少了计算开销和时间开销, 保障了 FactChain 执行和响应的效率.

算法 3. 真值验证算法.

输入: T, V, A;

输出: T^*, A.

1. $T^*=\{\}$;
2. **foreach** $t=(s,p,?,r)\in T$ **do**
3. t^*=置信度加权投票算法(t,V_t);
4. $Append\left(T^*,t^*\right)$;
5. $V_t^*=\{(ptcp,c)\mid(s,p,o)=t^*,v=(org,ptcp,s,p,o,c)\in V_t\}$;
6. $C=r+\sum_{(org,ptcp,s,p,o,c)\in V_t}c$;
7. A=垄断分红算法$\left(C,V_t^*,A\right)$;
8. **end**
9. **return** T^*, A;

FactChain 真值验证算法如算法 3 所示. 算法的主要时间开销花在对链上投票数据的访问上. 设全量投票数据为 $n=n_1+n_2+\ldots+n_m$, 其中, m 表示待验证的众包问题数量, $\{n_i,1\leqslant i\leqslant m\}$ 表示每个众包问题 t_i 对应的投票数量集合. 对于每一个众包问题 t_i: 置信度加权投票算法(算法 1)中评分计算的复杂度为 O(n_i), 选择真值的复杂度为 O(n_i); 垄断分红算法(算法 2)中阈值计算的复杂度为 O(n_i), 置信度分配的复杂度为 O(n_i). 对于每一

© 中国科学院软件研究所 http://www.jos.org.cn

个众包问题 t_i, 算法 3 串行调用置信度加权投票算法和垄断分红算法, 准备数据的复杂度为 O(n_i); 迭代处理所有众包问题的真值计算后, 算法 3 的复杂度合计为 O(n).

下面第 4.3.1 节中的多值真值验证和第 4.3.2 节中随时间变化知识的真值验证流程同样采用上述两阶段迭代式执行方式, 区别在于, 具体的真值验证算法和分红机制更加针对具体问题.

4.3 面向多值知识和随时间变化知识的扩展

FactChain 在融合多源冲突知识的真值验证问题上, 特别着重解决多值知识和随时间变化知识的验证. FactChain 支持参与人在发起众包任务请求时声明该知识是否是多值的或是随时间变化的, 详见图 2(a)中 Triple 状态概要.

4.3.1 多值真值验证

对于多源冲突知识的真值验证问题, 多数方法仅考虑每个目标问题存在一个真值的情况. 但是这种单真值的假设不总是成立的, 例如书籍和论文的作者、影视剧作品的演员等, 这些知识一般对应多个合理真值. 面对这种多值问题, 现有方法大多使用与单真值问题类似的方法来处理, 将数据源贡献的包含多个候选项的知识视作一个整体. 但是, 这种处理方式忽略了多值问题和单值问题之间的区别, 难以估计目标问题的最终结果数量, 也无法利用多个候选结果之间的兼容或互斥关系. 另一方面, 每个数据源的知识可能都是不完备的, 目标问题的真值可能需要由不同数据源给出的候选项组合得出, 而单真值处理方式仅能选择单个数据源贡献的知识, 无法生成由不同数据源中的部分候选项组合而成的输出.

针对多值问题, FactChain 采用一种基于加权投票的多真值验证方法. 首先估算目标问题对应的最终答案中互相兼容的结果数量, 然后依据估算结果数量确定最终结果集合. 将 m 个参与人提交的投票集合表示为 $V'=\{(ptcp_1,O_1=\{o_{1,1},o_{1,2},\ldots,o_{1,n_1}\},c_1),(ptcp_2,O_2=\{o_{2,1},o_{2,2},\ldots,o_{2,n_2}\},c_2),\ldots,(ptcp_m,O_m=\{o_{m,1},o_{m,2},\ldots,o_{m,n_m}\},c_m)\}$, 其中, $ptcp_i$ 和 c_i 分别表示第 i 个投票的参与人及其付出的置信度, n_i 表示该参与人投票的候选答案中包含的候选结果数量, $o_{i,j}$ 表示该参与人投票的第 j 个候选结果. 将多值问题的真值验证方法拆分为 3 个步骤, 结合表 3 的示例加以阐释, 该示例对应的真值验证问题是(“Bluetooth Application Programming with the Java APIs”,author,?).

表 3 多值问题真值验证示例

(a) 多值问题样例

voter	committed object	#candidate	confidence	**dividend**
$ptcp_1$	Kumar; Kline; Thompson	3	0.6	1.53
$ptcp_2$	Kumar; Kline; Thompson	3	0.8	2.04
$ptcp_3$	Kumar	1	0.5	0.42
$ptcp_4$	Kline	1	0.6	0.51
$ptcp_5$	Paul; Tim	2	0.8	0
$ptcp_6$	Bala; Paul; Thompson	3	0.8	0
$ptcp_7$	Tim; Bala; Thompson; Paul	4	0.4	0

(b) 结果数量评估

#candidate	$\sum confidence$
1	1.1
2	0.8
3	**2.2**
4	0.4

(c) 候选结果排序

ranked candidate	$\sum confidence$
Kline	1.07
Kumar	0.97
Thompson	0.83
…	…

(1) 结果数量评估. 相比于单真值事实验证问题, 多真值事实验证的一个难点在于结果的数量不确定. 出于这一考量, 首先根据参与人提交的候选答案中候选结果的数量和参与人的置信度进行加权投票估算最终答案中包含的结果数量, 通过公式(5)估算结果数量 l^*. 即以参与人提交的候选答案中的候选结果数量作为投票目标执行加权投票, 选择获得最高置信度的候选结果数量. 表 3(b)估算出该众包问题的结果数量应当是 3.

$$l^*=\underset{l\in\{n_i|i\in[1,m]\}}{\operatorname{argmax}}\left(\sum_{j\in[1,m]\wedge n_i=n_j}c_j\right) \tag{5}$$

(2) 候选结果排序. 确定候选结果数量之后, 以一种类似投资的方式根据公式(6)计算每个候选结果获得的总置信度, 即将参与人 $ptcp_i$ 投入的置信度均分给其支持的 n_i 个候选结果, 每个独立的候选结果获得所有提

© 中国科学院软件研究所 http://www.jos.org.cn

交的答案中包含其参与人的投资. 最后选择置信度得分排序前 l^* 的候选结果组成最终答案集合 O^*. 表 3(c)显示该问题最终答案集合为{Kline,Kumar,Thompson}.

$$score(o)=\sum_{i\in[1,m]\wedge o\in O_i}\frac{c_i}{n_i} \tag{6}$$

(3) 激励分配. 多真值问题相比于单真值问题更加复杂, 采用更加细粒度的激励分配策略. 部分参与人提交的候选答案仅为最终答案的子集, 这种投票也会对最终结果的确定起到积极作用, 应当予以激励. 部分参与人提交的候选答案虽然包含最终答案的子集, 但也包含其余干扰项, 这种投票对最终结果的确定造成了负面影响, 应当予以抑制. 基于上述认识, 采取公式(7)计算分配给参与人的分红.

$$dividend(ptcp_j)=\begin{cases}C\times\dfrac{\frac{|O_j|}{|O^*|}\times c_j}{\sum_{i\in[1,m]\wedge O_i\subseteq O^*}\frac{|O_i|}{|O^*|}\times c_i}, & O_j\subseteq O^*\\ 0, & \text{otherwise}\end{cases} \tag{7}$$

示例中具体分红方案展示在表 3(a)最右列.

在算法复杂度层面, 相比单真值问题的真值验证方法, 新增了结果数量评估流程. 设全量投票数据为 $n=n_1+n_2+\ldots+n_m$, 其中, m 表示待验证的众包问题数量, $\{n_i,1\leqslant i\leqslant m\}$ 表示每个众包问题 t_i 对应的投票数量集合. 在结果数量评估流程中, 对于每个众包问题 t_i, 根据投票数据中的候选结果数量进行一轮加权投票, 新增的复杂度为 O(n_i), 总的复杂度仍为 O(n).

4.3.2 随时间变化知识的真值验证

相比于传统真值验证问题的研究, FactChain 的真值验证更具时间上的动态性. 这种动态性一方面来自现实问题, 即知识图谱中的知识可能随时间变化而更新; 另一方面是由于 FactChain 的系统特性, 即参与人不断与系统动态交互. 现有的真值验证问题很少面临这种复杂的动态性. FactChain 在链上采用灵活的时序知识表达方式, 基于这种时序表达方式设计了一种针对具有复杂动态性的时序知识的真值验证算法. FactChain 的时序知识表达方式和时序知识真值验证算法为系统提供了充分的扩展空间, 可以便捷地适配交通传感器数据、智能穿戴设备数据等不间断生成且包含时间戳的信息.

根据关系的性质不同, 将知识图谱上的时间信息分为两种类型: 瞬时时间与持续时间. 出生、去世、获得学位这类知识一般对应瞬时时间, 就读、任职、居住这类知识一般对应持续时间. 直觉上, 带有瞬时时间信息的知识仅需要记录时间点, 而带有持续时间信息的知识则需要记录形如[开始时间,结束时间]的二元组. FactChain 对时间表示作了简化, 将上述两种类型的时序知识统一表示为形如(*s*,*p*,*o*,*ts*)的四元组. 对于带有瞬时时间信息的知识, *ts* 表示发生时间; 对于带有持续时间信息的知识, *ts* 表示最后生效时间. 这种表达方式上的简化是 FactChain 的增量式特性带来的, 链上能同时维护具有相同主语和谓语的三元组在不同时间段的版本, 可以通过该三元组前一最近版本的最后生效时间及当前版本的最后生效时间确定该时序知识的持续生效时间. 这种简化是有必要的, 一方面, 可减少链上存储的冗余; 另一方面, 这种统一的表示方式方便了真值验证流程.

对于时序知识请求 $t=(s,p,?)$, 假设其存在前一经验证版本 $t_{pre}=(s,p,o_{pre},ts_{pre})$, 来自 m 个参与人的未验证投票表示为 $V=\{(ptcp_1,o_1,c_1,ts_1),(ptcp_2,o_2,c_2,ts_2),\ldots,(ptcp_m,o_m,c_m,ts_m)\}$, 其中, 每个四元组由参与人、候选结果、置信度和时间戳组成. 时序知识真值验证的目的是消解未验证投票中的冲突, 得出当前版本的时序知识 (s,p,o_{cur},ts_{cur}). 简述 FactChain 中的时序知识真值验证算法如下.

1. 首先通过合并对同一候选结果的投票将投票重组为 k 个候选结果及其相对应投票的集合:

$$V=\{(o_1,\{(ptcp_{1,1},c_{1,1},ts_{1,1}),(ptcp_{1,2},c_{1,2},ts_{1,2}),\ldots,(ptcp_{1,n_1},c_{1,n_1},ts_{1,n_1})\}),(o_2,\{(ptcp_{2,1},c_{2,1},ts_{2,1}),(ptcp_{2,2},c_{2,2},ts_{2,2}),\ldots,(ptcp_{2,n_2},c_{2,n_2},ts_{2,n_2})\}),\ldots,(o_k,\{(ptcp_{k,1},c_{k,1},ts_{k,1}),(ptcp_{k,2},c_{k,2},ts_{k,2}),\ldots,(ptcp_{k,n_k},c_{k,n_k},ts_{k,n_k})\})\},$$

其中, o_i 和 n_i 分别表示第 i 个候选结果及支持其的参与人数量, $(ptcp_{i,j},c_{i,j},ts_{i,j})$ 为第 i 个候选结果的第 j

© 中国科学院软件研究所 http://www.jos.org.cn

个支持者、投入的置信度及其附加的时间戳.

2. 将前一经验证版本知识表示为 (o_{pre},r,P_{pre}), 其中, r 为众包请求发起者赋予该众包知识的激励额度, P_{pre} 表示该众包请求前一版本最终结果的支持者集合. 考虑到前一版本的支持者可能对新版本知识继续发起投票, 减去新版本参与人在上一版本的投票, 即更新前一版本的支持者集合为

$$P'_{pre}=P_{pre}-\{ptcp_{i,j}\mid i\in[1,k],j\in[1,n_i]\},$$

更新前一版本的置信度为 $C_{pre}=r\times\frac{|P'_{pre}|}{|P_{pre}|}$.

3. 将前一版本知识作为当前版本真值验证的一个候选结果, 得到当前所有候选结果及其置信度总和集合 $V'=\{(o_0=o_{pre},C_0=C_{pre}),(o_1,C_1),(o_2,C_2),\ldots,(o_k,C_k)\}$, 使用公式(8)计算每个候选结果获得的投票置信度总和. 这里, 每个候选结果获得的置信度总和 C_i 由两部分组成, 一部分是在该轮众包真值验证中获得的来自参与人的投票, 另一部分是真值验证算法对积极可信的参与人的额外偏好. 相比于陌生的参与人, 前一版本真值的支持者进行过一次有效投票, 可以被认定在专业性和可靠性两个层面均优于一般参与人, 因此在计算置信度时对其加以一定程度的偏好:

$$C_i=\sum_{j\in[1,n_i]}C_{i,j}+\frac{|(\{ptcp_{i,j}|j\in[1,n_i]\}\cap P_{pre})|}{|P_{pre}|}\times r,\ i\in[1,k] \tag{8}$$

4. 选择 $o^*=\mathrm{argmax}_{o^i\in\{o|(o,C)\in V'\}}C_i$, 即具有最高置信度总和的候选结果作为最终结果的宾语. 时间戳的生成需要考虑具体问题的类型, 即上文提到的时序信息属于瞬时时间还是持续时间, 选择众数时间 $ts^*=mode(\{ts_{*,j}\mid j\in[1,n_*]\})$ 作为瞬时时间知识的发生时间, 选择最大时间 $ts^*=\max(\{ts_{*,j}\mid j\in[1,n_*]\})$ 作为持续时间知识的最后生效时间. 至此生成该时序知识请求的新一版本 $(s,p,o_{cur}=o^*,ts_{cur}=ts^*)$.

5. 得出该时序真值验证问题的新一轮结果后, 发起新一轮激励分配, 将该轮新获得的投票置信度和众包激励的总和加权分配给支持最终结果的参与人. 每个参与人的分红按照公式(9)计算:

$$dividend(ptcp_{*,j})=\left(r+\sum_{i\in[1,k],p\in[1,n_i]}c_{i,p}\right)\times\frac{c_{*,j}}{\sum_{i\in[1,n_*]}c_{*,i}} \tag{9}$$

相比于传统真值验证算法, FactChain 定义了统一的时序知识表达方式, 算法层面采用结合前一版本结果计算下一版本真值的策略. 前一版本结果对下一版本真值验证流程的影响体现在两个方面. 其一, 将前一版本真值作为下一版本真值的候选结果, 即考虑到该时序知识的变更仅限于时间戳的更新而不改变三元组的宾语, 对应现实世界中譬如任期延长、活动延期等情况. 其二, 前一版本结果对下一版本真值验证的影响还体现在系统对上一版本最终结果支持者的偏好上, 在计算候选结果置信度总和时, 将参与新一轮投票且在前一轮投票中支持正确结果的参与人视作积极可信的参与人, 略微增加此类参与人的投票置信度.

考虑到时序知识可能不是互斥的, 而是在兼容的情况下不断增长. 例如请求关于诺贝尔文学奖获得者的知识 $(Nobel\_Prize\_in\_Literature,winner,?)$, 时间戳表示获奖者获奖的日期, 即诺贝尔奖的颁奖时间, 每年都会有新的获奖者, 因此该众包知识每年都会更新, 而新版本与旧版本之间不是互斥的, 即当新版本生成时, 旧知识并不失效. FactChain 在链上保存一条时序众包请求对应的所有经验证版本的知识, 以达成这种在动态迭代中向前兼容的特性.

一般真值验证算法很少考虑时序知识在时间线上的向前兼容性质, 在分析复杂度时, 本文将时序知识的后续真值推断视作一个新的众包问题. 设全量投票数据为 $n=n_1+n_2+\ldots+n_m$, 其中, m 表示待验证的众包问题数量, $\{n_i,1\leqslant i\leqslant m\}$ 表示每个众包问题 t_i 对应的投票数量集合. 对于一个存在前一版本真值的众包问题, 新增的操作为将前一版本真值作为新一轮投票的一个候选项, 新增的复杂度为 O(1), 总的复杂度仍为 O(n).

4.4 链下去中心化应用

为了保证单点崩溃容错和知识的开放共享, 链上知识被复制到区块链中每个分布式节点以供访问和操

© 中国科学院软件研究所 http://www.jos.org.cn

纵. 没有共享价值的数据和因隐私保护不允许流出组织外的知识应由组织在链下本地存储. 为了更好地融合链上和链下的知识以满足参与人查询和推理的需求, 链下实现去中心化应用程序(DApp), 主要负责结合链上链下知识响应参与人请求以及对参与人权限的管理. 从参与人角度来看, 链上与链下知识的结合是透明的. 部署在组织内部的去中心化应用考虑以下几点因素.

(1) 特定的参与人是否有权限发布众包请求、提交共享知识、查询相应知识?

(2) 如何确保由参与人提交的知识满足链上共享本体规范, 使得其余组织及参与人可理解?

(3) 如何正确、高效地结合链上和链下知识来响应组织内部参与人的查询请求?

结合对上述因素的思考, 将组织层面分布式应用的职能抽象为 3 项功能, 分别为参与人准入控制、模式转换和查询执行.

4.4.1 参与人准入控制

联盟链旨在为联盟成员提供安全可信的交流、共享、交易电子资产的平台. 出于跨组织知识图谱构建的体量和效率考量, FactChain 中每个组织通过保有区块链对等节点模块参与区块链网络, 组织内的参与人经由所属组织被链接进入区块链网络. 区块链上的智能合约负责自动化地响应组织的请求, 更新知识并管理参与人账户. 相应地, 部署在组织层面的分布式应用负责参与人准入控制. 通过链上智能合约提供的 Authorize 和 Deauthorize 等接口, 组织可以便捷地登记或注销参与人账户. 由于参与人的请求经由组织上传至区块链网络并调用智能合约, 并且区块链的回复也经由组织发送给参与人, 因此组织有权约束参与人对链上知识的访问和操纵权限.

以一个企业间协作共建知识图谱的场景为例分析. 根据企业员工的身份和职责不同, 对这些员工以不同程度开放 Request、Commit 以及 Query 命令的调用权限. 譬如, 访客没有访问链下私有数据的权限, 也不能通过 Request 命令向链上知识提交众包请求或者通过 Commit 命令向众包请求提交候选答案. 链上知识并不是凭空产生的, 而是由参与人贡献的, 其来源可以追溯到组织本地保存的链下知识图谱与参与人本身的私有知识. 如果确有数据隐私需求, 参与人准入控制模块同时负责对参与人提交的数据内容脱敏, 即审核用户上传的内容是否存在导致组织内部隐私泄露的风险.

需要特别说明的是, 参与人账户的额度在 FactChain 中兼具电子货币的激励属性和真值验证中置信度的作用. 为了起到切实的激励作用, 组织需要赋予组织内成员账户余额在实际生产生活中相应的价值. 同样地, 组织需要在分布式应用中设置参与人账户余额的下限, 当账户余额达到下限时, 组织即注销该参与人账户, 从而抑制恶意的参与人在 FactChain 网络中做出扰乱秩序的交互行为.

4.4.2 模式转换

由于多源知识自然的异构性, 不同组织本地的知识图谱可能使用不同的本体. 不同组织链下知识图谱中的本体间的不一致性会导致不同的参与人无法识别共指实体或共指关系, 给链上知识的融合造成阻碍. 组织层面的去中心化应用负责模式转换功能, 消解不同组织使用的不一致本体造成的异构性.

在 FatcChain 正式运行接受参与人请求之前, 参与知识图谱协作共建的各组织联合创建一个统一的本体作为链上知识图谱使用的全局本体. 每个组织在本地维护一个全局共享本体和本地私有本体之间的映射关系. 在当前的 FactChain 系统中, 这个映射是由每个组织人工构建的. 通过全局共享本体和本地私有本体之间的映射关系, 组织将参与人上传的基于私有本体表示的知识转换为基于统一本体表示的知识并提交上链, 也将智能合约返回的基于统一本体表示的知识转换为私有本体下的知识回传给发起请求的参与人. 在模式转换功能的保障下, 各组织内的参与人可以无视组织间私有本体的异构性, 开放、透明地共享知识.

4.4.3 查询执行

作为连接组织内参与人和区块链的桥梁, 组织需要响应组织内参与人对知识的查询甚至更复杂的知识推理请求. 一方面, 组织对参与人提交的查询请求进行模式转换、调用相应的智能合约查询链上知识, 并将智能合约返回的基于链上全局本体表示的知识转换为基于本地私有本体表示的知识. 另一方面, 组织查询自己本地保存的链下知识图谱. 最终, 组织将链上知识和链下知识结合将结果返回给发起查询请求的参与人.

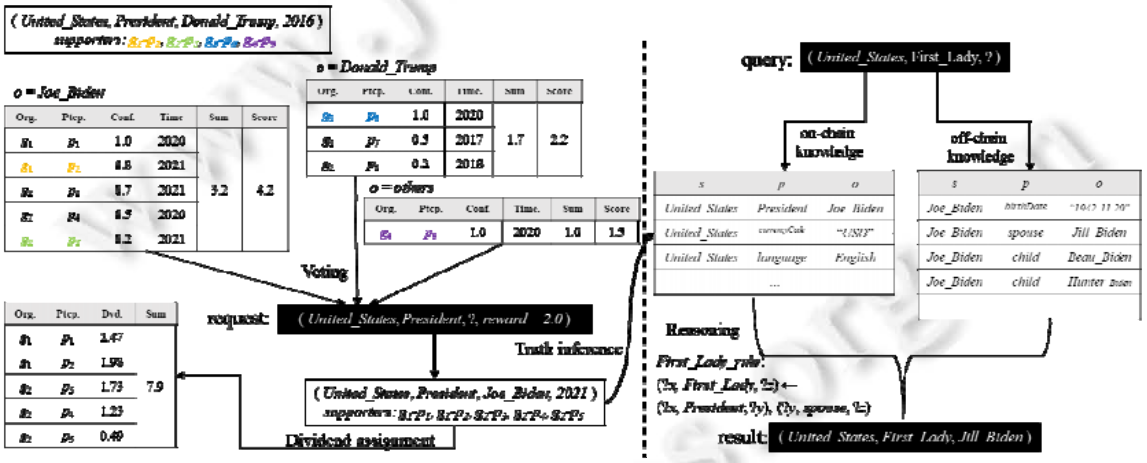
© 中国科学院软件研究所 http://www.jos.org.cn

一般来说, 区块链系统提供基于 NoSQL 的数据访问方式, 例如通过键值对的方式访问数据. 在 FactChain 中, 参与人可以通过声明具体的头实体和关系来查询对应的三元组. 为了支持对链上知识更加方便和多样化的访问, FactChain 通过智能合约实现了对具有同一个头实体的所有三元组的访问和对一个中心实体的多跳知识的查询. 值得注意的是, 还可以通过智能合约和在链下分布式应用定制不同类型索引的方式来支持更加复杂的数据访问.

5 应用演示

FactChain 定位为一个通过跨组织的知识共享与融合来协作创建知识图谱的智慧信息系统, 作为后续以数据为核心的应用的支撑与保障. 通过丰富组织层面去中心化应用实现了基于跨组织知识融合的、结合链上与链下知识的推理.

图 4(a)展示了 FactChain 进行众包知识融合的主要流程. 设每个组织内参与人初始账户余额为 100. 将触发真值验证流程的置信度总和阈值设置为 5.0. 链上存在时序知识众包请求 (*United _States*,*President*,?), 即请求当前美国总统的时序知识. 链上知识图谱中存在已验证的前一版本知识(*United_States*,*President*, *Donald_Trump*,2016), 即 2016 年美国总统为 Donald Trump. 在图 4(a)所示的样例中, 5 个参与人为“Joe_Biden”投票, 置信度总和为 3.2, 其中两个参与人是上一版本真值的支持者, 获得置信度加成, “Joe_Biden”得到的总分为 4.2. 3 个参与人提交的候选结果为“Donald_Trump”, 置信度总和为 1.7, 得分为 2.2, 其余不表. 上述参与人提交的投票置信度总和达到了预设的阈值, FactChain 在链上通过执行预定义的智能合约运行真值验证, 候选值“Joe_Biden”因其具有最高得分 4.2 而被选择作为当前版本真值. 紧接着, 总额为 7.9 的置信度按照投票加权分配至当前版本真值“Joe_Biden”的支持者. 链上生成该时序知识当前版本(*United_States*,*President*,*Joe_Biden*,2021).



(a) 知识融合 (b) 知识推理

图 4 结合知识融合与推理的应用演示

图 4(b)展示了结合链上与链下知识的查询推理应用. FactChain 中的推理功能为基于 Apache Jena[28]实现, 通过在知识图谱上执行预定义的霍恩推理规则从已有知识推断出新知识. 例如, 公式(10)展示了一个简单的霍恩规则.

$$First\_Lady\_rule:(?x,First\_Lady,?z)\leftarrow(?x,President,?y),(?y,spouse,?z) \tag{10}$$

上述规则的含义是第一夫人一般指总统的妻子. 如果知识图谱中的知识满足规则的右侧(前提)且知识图谱中不存在该规则的左侧(结论), 则该规则被触发且生成相应的结论. 为了在保障查询效率的前提下丰富链下知识图谱, FactChain 采用一种目标驱动的反向链推理(backward-chaining)范式. 当组织接收到一个参与人的查询请求时, 首先在链下知识图谱上执行查询. 如果在组织本地的知识图谱上查询不到目标结果, 组织则通

© 中国科学院软件研究所 http://www.jos.org.cn

过调用智能合约抓取链上知识中以该查询三元组主语为中心的两跳以内的三元组, 基于模式转换模块将相关的链上知识转换并存入本地图谱. 基于预定义的推理规则在扩充后的链下知识图谱上执行推理, 将推理生成的结论作为新生成的知识添加进链下知识图谱. 最终, 组织再次在链下知识图谱上执行查询语句, 返回查询结果.

在图 4(b)的示例中, 参与人提交了对美国第一夫人的查询, 但是链上与该组织内部的知识图谱中均不存在该知识. 部署在该组织的分布式应用从 FactChain 的链上知识图谱中抓取所有和美国相关的知识, 包括总统、语言、货币等等. 其中, 总统信息对于该查询是有意义的, 因为该知识出现在推理第一夫人的规则的前提中. 链上知识图谱中关于美国的三元组被添加进链下知识图谱, 现在从该链下知识中可以得知美国现任总统为 Joe Biden、其妻子为 Jill Biden, 公式(10)的前提被满足, 推理时触发该条规则并生成美国的第一夫人为 Jill Biden 的相关三元组作为查询结果返回.

在上述结合链上与链下知识的推理过程中, 保存在组织链下的知识图谱经历了两次扩充, 第 1 次是将链上与查询相关的三元组抓取到链下, 即使用跨组织融合的知识丰富组织内知识图谱; 第 2 次是将根据预定义规则推理得出的三元组添加进知识图谱, 即基于规则丰富知识图谱本身. FactChain 通过上述两次链下知识图谱的扩充, 达成了知识再生产的目标.

6 实 验

本节设计实验以验证 FactChain 的可用性与实用性. 首先通过测试在不同区块链网络配置下 FactChain 的性能表现, 选取最优的区块链网络配置, 然后评估 FactChain 实现的真值验证算法在真实数据集上的效果.

6.1 数据集

FactChain 着眼于处理跨组织知识融合场景中的多源冲突知识的真值验证问题, 本文将真值验证真实数据集处理成 FactChain 众包构建知识图谱的请求序列, 通过模拟该请求序列的执行来评估系统的可用性和有效性. 选用真值验证问题常用的 Book-Author 数据集[29], 该数据集收集自 www.abebooks.com, 包含出自 894 个在线书店的关于 1 265 本计算机科学书籍的销售记录, 共计 34 031 条数据. 将数据集转换为“书籍-作者”的众包知识图谱构建的请求序列, 对应在线知识图谱中形如 ($book\_isbn$,$book\_author$,$author\_list$) 的三元组, 即每本书籍对应的作者列表. 实验中的请求序列中共包含 1 265 个 Request 请求、34 031 个 Commit 请求以及 1 265 个 Query 请求. 该数据集中的真值验证问题存在多值现象, 可以验证 FactChain 处理多值真值验证问题的能力. 由于现存时序真值验证数据集不适合知识图谱构建的场景, 实验中未涉及时序知识真值验证. FactChain 会在开放运行过程中不断收集并融合时序知识, 以便构造后续可用的数据集.

6.2 众包构建知识图谱性能测试

首先简要描述 FactChain 的运行环境和系统配置. FactChain 部署在 7 台服务器上, 每台服务器为 8 核 CPU、16 GB 内存及 Ubuntu 18.04 操作系统. 基于常用的开源联盟链框架 Hyperledger Fabric 2.2 版搭建了由 4 个对等节点和 3 个排序节点组成的区块链网络. 区块链网络中的对等节点负责模拟执行智能合约、为其他节点的请求背书, 排序节点根据背书策略对执行智能合约生成的读写集排序. 选用 Hyperledger Fabric 中可插拔的 Raft 作为共识机制, 以实现单点崩溃容错性. Raft 采用一种动态选举主节点的主从模型, 只要网络中过半节点正常运行, 系统就能达到最终一致性.

衡量区块链网络性能的常用指标有 TPS (transactions per second)和 Latency. TPS 为区块链网络平均每秒正确执行的事务数量, Latency 为从事务提交到智能合约执行结果写入链上的平均等待时间. 从系统配置层面影响区块链性能的主要有 MaxMessageCount 和 BatchTimeout 两个参数, 其中, MaxMessageCount 设置每个区块最多包含的事务数量, BatchTimeout 设置最大出块时延. 当前累计事务数量达到 MaxMessageCount 或者上一区块生成超过 BatchTimeout 时间, 即将当前所有事务打包生成新区块.

使用开源区块链测试工具 Hyperledger Caliper 测试 FactChain 性能. 为了模拟 FactChain 在实际运行时的

© 中国科学院软件研究所 http://www.jos.org.cn

真实性能指标, 将从真值验证数据集构造的共计 36 561 条知识图谱构建请求序列随机打乱, 以平均每秒 100 条事务请求的速率提交至系统中, 在不同的区块链网络配置组合下进行 3 次实验计算均值和标准差. 表 4 和表 5 分别记录了 FactChain 在不同区块链网络配置下 TPS 和 Latency 指标的变化情况. 从表中可以看出, 当 MaxMessageCount 设置为 1 时, 每条事务单独打包成一个区块, 系统的通信开销大, 单个区块的存储有较大冗余, 呈现出低 TPS 和高 Latency. 随着 MaxMessageCount 的增大, 区块内包含的事务数量也增大, 相对通信开销减少, 处理一个区块的计算开销增大, 平均到每条事务的计算开销减少, 系统的 TPS 缓慢增大趋于稳定, Latency 缓慢增大. 随着 BatchTimeout 的增大, 出块速率减缓, Latency 缓慢增大. 从提高吞吐率、降低平均时延、节约计算开销和通信开销等角度综合考虑, 最终选择的区块链网络配置为 MaxMessageCount=32 且 *BatchTimeout*=1.

表 4 TPS 随 MaxMessageCount 和 BatchTimeout 变化

TPS (SD)		MaxMessageCount				
		1	8	16	32	64
BatchTimeout (s)	0.2	69.63 (1.00)	115.13 (1.60)	121.17 (1.96)	124.57 (2.10)	124.07 (1.56)
	1	68.73 (0.80)	106.03 (1.56)	111.70 (0.95)	117.33 (0.86)	122.23 (1.70)
	2	69.40 (0.90)	98.53 (1.51)	105.73 (0.85)	104.53 (0.59)	110.00 (1.37)
	3	69.23 (0.72)	91.97 (0.12)	95.47 (1.11)	99.27 (0.76)	101.97 (1.10)

表 5 Latency (s)随 MaxMessageCount 和 BatchTimeout 变化

Latency (SD)		MaxMessageCount				
		1	8	16	32	64
BatchTimeout (s)	0.2	2.94 (0.06)	0.15 (0.01)	0.17 (0.01)	0.23 (0.01)	0.23 (0.01)
	1	3.03 (0.04)	0.15 (0.01)	0.18 (0.01)	0.25 (0.01)	0.42 (0.01)
	2	3.39 (0.10)	0.15 (0.01)	0.21 (0.01)	0.30 (0.01)	0.55 (0.02)
	3	2.24 (0.21)	0.16 (0.01)	0.23 (0.02)	0.32 (0.02)	0.55 (0.04)

6.3 真值验证效果评估

考虑到真值验证选择的结果可能是正确答案的一部分, 采用广泛使用的 Jaccard 相似度作为评价指标. 数据集中包含 100 条人工标注出正确答案的测试集, 将测试集上所有真值验证的预测值与正确答案的相似度加和作为评价分数. 选择区块链系统中最常用的结合群体智慧的 MV (majority voting)算法和经典的迭代式真值验证算法 PooledInvestment[30]作为对比. 表 6 展示了两种对比方法和 FactChain 在 Book-Author 数据集上的真值验证效果, 其中, Hits 表示完全正确的结果数量, Sim100 表示 100 条测试数据上预测值和真值的相似度之和. 可以看出, FactChain 的真值验证算法无论是在完全命中数量还是在相似度上都明显优于两个对比方法.

表 6 最右列展示了 FactChain 和两种对比方法的算法复杂度. 如前文所述, 真值验证问题的主要开销是对投票数据的访问, 本文中设投票数据总量为 n. 对比方法 MV 仅进行一次多数投票, 即对全量数据进行一次访问. 对比方法 PooledInvestment 进行 k 次迭代直至收敛, 每次迭代在全量数据上进行真值推断和置信度分配. 而 FactChain 的真值验证算法也仅访问一次全量数据, 具体参见第 4.2.3 节.

表 6 真值验证结果评估

	Hits	Sim100	平均复杂度
MV	56	74.4	O(n)
PooledInvestment	59	75.7	O(kn)
FactChain	69	84.6	O(n)

7 总结与展望

本文研究了使用区块链来实现知识图谱的构建与管理, 提出了基于区块链的众包知识融合系统 FactChain. FactChain 着力于解决知识融合中多源冲突知识的真值验证问题, 设计了适应区块链架构的置信度加权投票算法和垄断分红算法. 此外, FactChain 面向事实验证现实问题, 对多值知识和时序知识的真值验证

© 中国科学院软件研究所 http://www.jos.org.cn

给出了有效方法. 作为创建去中心化知识管理新生态的初步尝试, 本文展示了 FactChain 在知识融合的基础上结合链上与链下知识的查询与推理功能. 本文还通过真实数据集度量了系统的可用性和有效性.

未来, FactChain 将以特定领域大规模应用落地为目标, 发展更具智能的实用信息系统. 在链下去中心化应用中, FactChain 的模式转换模块目前使用预定义的链上本体与本地本体映射来实现链上知识本地化与链下知识全局化. 下一步将尝试采用自动化本体映射技术, 减少系统所需的人工干预. 此外, FactChain 结合链上与链下知识的推理目前是基于预定义霍恩规则的后向链推理, 未来会考虑更丰富而高效的推理方式.

References:

[1] Berners-Lee T, Hendler J, Lassila O. The semantic Web. Scientific American (JSTOR), 2001, 284(5): 34–43.

[2] Auer S, Bizer C, Kobilarov G, *et al*. DBpedia: A nucleus for a Web of open data. In: The Semantic Web. Springer, 2007. 722−735.

[3] Vrandečić D, Krötzsch M. Wikidata: A free collaborative knowledgebase. Communications of the ACM, 2014, 57(10): 78−85.

[4] Wu W, Li H, Wang H, *et al*. Probase: A probabilistic taxonomy for text understanding. In: Proc. of the 2012 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). ACM, 2012. 481−492.

[5] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review, 2008, 21260.

[6] Yuan Y, Wang F. Blockchain: The state of the art and future trends. Acta Automatica Sinica, 2016, 42(4): 481−494 (in Chinese with English abstract).

[7] Notheisen B, Willrich S, Diez M, *et al*. Requirement-driven taxonomy development—A classification of blockchain technologies for securities post-trading. In: Proc. of the 52nd Hawaii Int'l Conf. on System Sciences (HICSS). IEEE, 2019. 1−10.

[8] Su X, Liu Y, Choi C. A blockchain-based P2P transaction method and sensitive data encoding for e-commerce transactions. IEEE Consumer Electronics Magazine, 2020, 9(4): 56−66.

[9] Reyna A, Martín C, Chen J, *et al*. On blockchain and its integration with IoT. Challenges and opportunities. Future Generation Computer Systems, 2018, 88: 173−190.

[10] Han X, Liu Y. Research on the consensus mechanisms of blockchain technology. Netinfo Security, 2017, 17(9): 147−152 (in Chinese with English abstract).

[11] Ouyang L, Wang S, Yuan Y, *et al*. Smart contracts: Architecture and research progresses. Acta Automatica Sinica, 2019, 45(3): 445−457 (in Chinese with English abstract).

[12] Wood G. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 2014, 151(2014): 1−32.

[13] Androulaki E, Barger A, Bortnikov V, *et al*. Hyperledger fabric: A distributed operating system for permissioned blockchains. In: Proc. of the 13th European Conf. on Computer Systems (EuroSys). ACM, 2018. 1−15.

[14] Naim BA, Klas W. Knowledge graph-enhanced blockchains by integrating a graph-data service-layer. In: Proc. of the 6th Int'l Conf. on Internet of Things: Systems, Management and Security (IOTSMS). IEEE, 2019. 420–427.

[15] Hector UR, Boris CL. BLONDiE: Blockchain ontology with dynamic extensibility. CoRR, abs/2008.09518, 2020.

[16] Zhu Y, Zhang Z, Jin C, *et al*. SEBDB: Semantics empowered blockchain database. In: Proc. of the 35th Int'l Conf. on Data Engineering (ICDE). IEEE, 2019. 1820−1831.

[17] Fill HG, Härer F. Knowledge blockchains: Applying blockchain technologies to enterprise modeling. In: Proc. of the 51st Hawaii Int'l Conf. on System Sciences (HICSS). IEEE, 2018. 1−10.

[18] Sopek M, Gradzki P, Kosowski W, *et al*. GraphChain: A distributed database with explicit semantics and chained RDF graphs. In: Proc. of the 27th Int'l World Wide Web Conf. (WWW). ACM, 2018. 1171−1178.

[19] Lin X, Li J, Wu J, *et al*. Making knowledge tradable in edge-AI enabled IoT: A consortium blockchain-based efficient and incentive approach. IEEE Trans. on Industrial Informatics, 2019, 15(12): 6367−6378.

[20] Emaldi M, Zabaleta K, López-de-Ipiña D. AUDABLOK: Engaging citizens in open data refinement through blockchain. In: Proc. of the 13th Int'l Conf. on Ubiquitous Computing and Ambient Intelligence (UCAmI). MDPI, 2019, 31(1): 52.

[21] Jaroucheh Z, Alissa M, Buchanan WJ, *et al*. TRUSTD: Combat fake content using blockchain and collective signature technologies. In: Proc. of the 44th Annual Int'l Computer Software and Applications Conf. (COMPSAC). IEEE, 2020. 1235−1240.

[22] Chen H, Hu N, Qi G, *et al*. OpenKG chain: A blockchain infrastructure for Open Knowledge Graphs. Data Intelligence, 2021, 3(2): 205−227.

© 中国科学院软件研究所 http://www.jos.org.cn

[23] Li Y, Gao J, Meng C, *et al*. A survey on truth discovery. ACM SIGKDD Explorations Newsletter, 2015, 17(2): 1−16.

[24] Lavi R, Sattath O, Zohar A. Redesigning Bitcoin's fee market. In: Proc. of the 28th Int'l World Wide Web Conf. (WWW). ACM, 2019. 2950−2956.

[25] Goldberg AV, Hartline JD, Karlin AR, *et al*. Competitive auctions. Games and Economic Behavior, 2006, 55(2): 242−269.

[26] Yao AC. An incentive analysis of some Bitcoin fee designs. In: Proc. of the 47th Int'l Colloquium on Automata, Languages, and Programming (ICALP). LIPIcs, 2020, 168(1): 1−12.

[27] Wang W, Jiang J, An B, *et al*. Toward efficient team formation for crowdsourcing in noncooperative social networks. IEEE Trans. on Cybernetics, 2016, 47(12): 4208−4222.

[28] Carroll JJ, Dickinson I, Dollin C, *et al*. Jena: Implementing the semantic Web recommendations. In: Proc. of the 13th Int'l Conf. on World Wide Web—Alternate Track Papers & Posters (WWW). ACM, 2004. 74−83.

[29] Yin X, Han J, Yu PS. Truth discovery with multiple conflicting information providers on the Web. IEEE Trans. on Knowledge and Data Engineering, 2008, 20(6): 796−808.

[30] Pasternack J, Roth D. Knowing what to believe (when you already know something). In: Proc. of the 23rd Int'l Conf. on Computational Linguistics (COLING). ACL, 2010. 877−885.

附中文参考文献:

[6] 袁勇, 王飞跃. 区块链技术发展现状与展望. 自动化学报, 2016, 42(4): 481−494.

[10] 韩璇, 刘亚敏. 区块链技术中的共识机制研究. 信息网络安全, 2017, 17(9): 147−152.

[11] 欧阳丽炜, 王帅, 袁勇, 等. 智能合约: 架构及进展. 自动化学报, 2019, 45(3): 445−457.

朱向荣(1998－), 男, 博士生, CCF 学生会员, 主要研究领域为知识融合.

吴鸿祜(1998－), 男, 硕士生, CCF 学生会员, 主要研究领域为区块链数据管理.

胡伟(1982－), 男, 博士, 副教授, 博士生导师, CCF 高级会员, 主要研究领域为知识图谱, 数据集成, 智能软件.

© 中国科学院软件研究所 http://www.jos.org.cn