An RDF Data Set Quality Assessment Mechanism for Decentralized Systems Postprint

Authors: Huang, Li, Liu, Zhenzhen, Xu, Fangfang, Gu, Jinguang, Gu, Jinguang

Date: 2022-11-18T00:00:00+00:00

Abstract

With the rapid growth of the linked data on the Web, the quality assessment of the RDF data set becomes particularly important, especially for the quality and accessibility of the linked data. In most cases, RDF data sets are shared online, leading to a high maintenance cost for the quality assessment. This also potentially pollutes Internet data. Recently blockchain technology has shown the potential in many applications. Using the blockchain storage quality assessment results can reduce the centralization of the authority, and the quality assessment results have characteristics such as non-tampering. To this end, we propose an RDF data quality assessment model in a decentralized environment, pointing out a new dimension of RDF data quality. We use the blockchain to record the data quality assessment results and design a detailed update strategy for the quality assessment results. We have implemented a system DCQA to test and verify the feasibility of the quality assessment model. The proposed method can provide users with better cost-effective results when knowledge is independently protected.

Full Text

Preamble

RESEARCH PAPER
An RDF Data Set Quality Assessment Mechanism for Decentralized Systems

Li Huang[12], Zhenzhen Liu[12], Fangfang Xu[12] & Jinguang Gu[123]†
[1]Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, Wuhan 430065, China
[2]College of Computer Science and Technology, Wuhan University of Science and

Technology, Wuhan 430065, China
[3]Key Laboratory of Rich-media Knowledge Organization and Service of Digital Publishing Content, State Administration of Radio and Television, Beijing 100038, China

Keywords: Decentralization; Quality assessment; Blockchain; RDF data set

Citation: L. Huang, Z. Liu, F. Xu & J. Gu. An RDF data set quality assessment mechanism for decentralized systems. *Data Intelligence* 2(2020), 529–553. doi: 10.1162/dint_a_{00059}

Received: December 27, 2018; **Revised:** October 25, 2019; **Accepted:** March 20, 2020

Abstract

With the rapid growth of linked data on the Web, the quality assessment of RDF data sets has become particularly important, especially concerning the quality and accessibility of linked data. In most cases, RDF data sets are shared online, leading to high maintenance costs for quality assessment and potential data pollution across the Internet. Recently, blockchain technology has demonstrated potential across many applications. Using blockchain to store quality assessment results can reduce reliance on centralized authorities while providing characteristics such as immutability. To this end, we propose an RDF data quality assessment model for decentralized environments, introducing a new dimension to RDF data quality assessment.

We utilize blockchain to record data quality assessment results and design a detailed update strategy for these results. We have implemented a system called DCQA to test and verify the feasibility of our quality assessment model. The proposed method can provide users with more cost-effective results while ensuring independent protection of knowledge.

1. Introduction

Current decentralized systems have experienced explosive growth. A decentralized network structure is a network architecture that grants permissions to member nodes without a central authority. Its structural characteristics are: (1) it has many nodes; (2) there is no hierarchical relationship between nodes; (3) each node can run independently without interference from other nodes; and (4) nodes are interconnected. Compared with centralized structures, decentralized architectures reduce dependence on central nodes and enhance the security and robustness of the entire system. Phenomenal decentralized products such as Bitcoin [**?**] proposed by Satoshi Nakamoto, and the blockchain technology [**?**] it employs, represent a decentralized ledger. Presently, blockchain technology is gradually separating from Bitcoin and, as an independent technology, is now widely used in many fields [**?**], demonstrating excellent performance in finance, supply chain, and insurance.

The core content of the third-generation Internet—semantic Web technology [?]—redefines resource description methods, leading to the emergence of the Resource Description Framework (RDF) [?, ?]. RDF aims to describe resources on the network and fundamentally solve problems such as data integration difficulties and machine understanding challenges caused by different data description formats on the Internet. With the continuous increase in decentralized systems using RDF data sets, the quality of original RDF data sets has become increasingly important and significantly affects system performance. Many fields use RDF data structures for transaction processing. For example, in the medical domain, various hospitals provide RDF data to form a comprehensive pharmaceutical knowledge graph for the convenience of users or physicians. Different hospitals may have identical drug information or attributes, while some hospitals possess unique information or attributes. Obtaining more cost-effective and accurate query results when users search for drug information becomes a key challenge. Thus, RDF quality assessment across different domains is crucial. The emergence and development of blockchain have provided new ideas and inspiration for addressing RDF data quality assessment issues. This paper uses blockchain to store quality assessment results, which can reduce the centralization of authoritative institutions while ensuring that quality assessment results have characteristics such as immutability. In decentralized systems, RDF data quality assessment introduces multiple new dimensions such as completeness and uniqueness.

We propose an RDF data set quality assessment mechanism for decentralized systems and discuss how to implement partial dimensions of RDF data quality assessment under the condition that the knowledge of each node in the blockchain is independently protected. In particular, we have implemented a system to test and verify the feasibility of the quality assessment model, which we named DCQA.

2. Related Work

The quality assessment of RDF data sets has received widespread attention, with many researchers conducting in-depth investigations into RDF quality evaluation [?, ?, ?]. Numerous studies on RDF data set quality assessment have approached the problem from different perspectives, summarizing multiple dimensions of RDF across production, usage, and maintenance phases. Furthermore, with the rapid development of RDF, new dimensions continue to emerge. Assaf and Senart [?] summarized five types of linked data quality assessment principles: quality of data source, quality of raw data, quality of semantic conversion, quality of the linking process, and global quality. These five principles describe the content of quality assessment work at each stage from a data management process perspective. The principles at each stage include several assessment standards, which constitute the dimensions of quality assessment in this paper. Zaveri et al. [?] summarized more than ten linked data quality assessment papers, introduced existing quality assessment methods in detail, and divided quality as-

sessment dimensions into six categories from different assessment aspects. They also proposed a unified description of terms, dimensions, and measurement indicators regarding data quality. Using flight information as an example, they demonstrated the feasibility of their scheme through close integration with quality assessment in practice. Bizer and Cyganiak [**?**] divided quality dimensions into three categories according to the type of information used: content-based (referring to the content of the information itself), context-based (referring to contextual information where the information is declared), and level-based (referring to the level of the data itself or the information provider). With the development of RDF data quality assessment studies, research on decentralized systems using blockchain technology is increasing.

Governments worldwide have pursued policies to promote the adoption and development of blockchain. For example, the Canadian government intends to use the Ethereum blockchain to track and record government donation information; the British government released a special report on blockchain applications in government work and finance [**?**]. The Indian government has launched a blockchain ecosystem in cooperation with fund company Covalentfund. Blockchain technology research has become one of the hot research fields.

In a decentralized system, the RDF knowledge of each node is independent and protected. Therefore, the quality assessment of RDF data differs significantly from centralized distributed systems, with many new dimensions and updated methods emerging in recent years. In quality assessment of RDF data combined with blockchain, we should not only pay attention to the data quality of the RDF data set itself but also to quality issues caused by interactions between RDF data sets. Previous studies on quality assessment were based on single data sets. In response to these problems, this paper proposes an RDF data set quality assessment mechanism for decentralized systems, aiming to provide users with better services in a decentralized environment.

3. Design of Quality Assessment Model

In decentralized systems, node quality consists of two parts: node service quality and node data quality. Node service quality refers to a node' s ability to effectively provide services, which is generally affected by the physical factors of the node itself. Node data quality measures the quality of service provided by a node. Since node service quality is limited by physical constraints and does not change significantly, this paper focuses on node data quality.

3.1 RDF Inspection Report

An RDF data set has certain quality factors including the number of blank nodes, data redundancy, and accessibility of Uniform Resource Identifiers (URIs). Combining these three RDF data quality issues with the average number of subject attributes in the RDF data set, we designed and implemented an inspection report model to quantify RDF data quality. For RDF data

attribute symbols used in this section, please refer to Table 1 .

Table 1. RDF data set basic attributes

Attribute description	Symbol
Blank node number	Blank(data)
Number of subjects	S(data)
The number of unique subjects	US(data)
Number of predicates	P(data)
The number of unique predicates	UP(data)
The number of objects	O(data)
The number of unique objects	UO(data)
The number of triples	SPO(data)
The number of unique triples	USPO(data)
URI accessibility	URI(data)

RDF data set redundancy calculation equation:

$$\text{Redundancy(data)} = 1 - \frac{\text{USPO(data)}}{\text{SPO(data)}}$$

Equation for calculating the average number of subjects in an RDF data set:

$$\frac{\text{SPO(data)}}{\text{US(data)}}$$

The average number of attributes in an RDF data set indicates the description level of a subject in a data set, where a larger value indicates that the data set uses more triples to describe the subject. Increased subject attributes can make knowledge more complete, so the number of RDF average properties is proportional to the quality of the RDF data set.

The RDF inspection report model is given below:

$$\text{QRDF(data)} = k_1 \cdot \text{Redundancy(data)} + k_2 \cdot \frac{\text{Blank(data)}}{\text{SPO(data)}} + k_3 \cdot \text{URI(data)}$$

Sample access to the URIs in the RDF data set, and then URI(data) in Equation (3) is the ratio between the number of accessible URIs and the total number of random checks. k_1, k_2, k_3, and k_4 are positive weight coefficients that can be adjusted according to different systems.

3.2 Verifiability

Verifiability refers to obtaining identical query results from multiple identical queries. Frequent data changes can cause users to distrust a node' s data, yet data updates and modifications are necessary. Therefore, verifiability varies as the data set changes. This paper sets the verifiability granularity to the log level, meaning each query generates logs. The latest record is compared with the previous record to obtain accuracy and error rates, and the difference value represents the verifiability result of the query. Node verifiability is calculated as follows:

$$\text{Verifiability(data)} = \text{ErrorRate(data)} + \text{CorrectRate(data)}$$

According to Equation (4), when a node updates data, its verifiability decreases significantly, resulting in declining data quality. However, as queries increase, verifiability gradually improves.

The granularity of verifiability is log-level. Although not refined to the entity level, this approach is more practical. For example, an entity in node A has five attributes, and new attributes need to be added. Due to low user demand, the entity change has minimal effect on the entire knowledge graph, and its verifiability fluctuations are not substantial. If entity-level granularity were adopted, it would consume more space without obvious benefits. We design logs for each query, with the format shown in Figure 1 [Figure 1: see original paper].

Figure 1. Query log structure

The query hash records the hash of each query result for comparison with the next hash value. The transaction ID corresponds to the query transaction ID and is used to calculate transaction information. The last part of the query log is the transaction time. For example, a log list of node A is shown in Table 2 .

Table 2. Query log instance

Transaction ID	Query hash	Query result hash	Query delay
MrEzFxxu8ECd5R1-BoJLNjhWd4GcUNGV	edy8vvAmItbgD2[illegible]jcr0jZhWI14C6		200ms
MrEzFxxu8ECd5R1-BoJLNjhWd4GcUNGV	edy8vvAmItbgD2[illegible]jcr0jZhWI14C6		200ms
MrEzFxxu8ECd5R1-nJbPpd2WP6i78GcG	edy8vvAmItbgD2[illegible]jcr0jZhWI14C6		300ms
MrEzFxxu8ECd5R1-nJbPpd2WP6i78GcG	edy8vvAmItbgD2[illegible]jcr0jZhWI14C6		600ms

The verifiability of node A is 1. The results of log 2 and log 3 in the log list are different, indicating low verifiability, which leads to declining data quality. The

verifiability of a stable node is $D_{\log}(\text{data})$, which is the number of unique query hashes. If this number is smaller than this value, the verifiability of the node decreases (see Equation (5)).

3.3 Completeness, Relevance and Uniqueness

To explain the calculation of completeness and other dimensions, Table 3 provides a list of symbols and their meanings.

Table 3. Partial symbols and their significance in model calculation

Symbol	Description
$P(\text{data}, S_i)$	The number of predicates in the Data with the subject of S_i
$UP(\text{data}, S_i)$	The number of unique predicates in Data
$P(S_i)$	The number of predicates with the subject of S_i in the entire knowledge graph
$BP(S_i)$	The number of predicates and BP intersections in the subject of S_i in Data

Completeness, relevance, and uniqueness are new quality assessment indicators that ensure the knowledge of each node in the decentralized system has its own intellectual property rights. Each node owns a portion of the entire system knowledge graph, so the proportion of each part in all attributes of the entire knowledge graph affects the degree to which the node contributes to the overall knowledge graph.

If node A has an entity S_i that does not exist in other nodes, and users have greater demand for S_i, then the data set in node A contributes significantly to the entire knowledge graph. In contrast, if each node owns the same entity, that entity' s contribution in each node is small.

The equation for calculating the completeness of entity S_i in the data set data is:

$$\text{Completeness}(\text{data}, S_i) = \frac{P(\text{data}, S_i)}{P(S_i)}$$

According to Equation (6), the sum of completeness across all nodes is not 1 because attributes of the same subject in different nodes may be duplicated. Relevance is the sum of the proportion of repeated attributes calculated by entity S_i in different nodes, representing the similarity between node data sets. Its calculation equation is:

$$\text{Relationship}(\text{data}, S_i) = \sum_{S_i \in \text{Data}} \sum_{\text{Eachothernode}} \frac{BP(S_i)}{P(S_i)}$$

In contrast, uniqueness refers to the degree to which a node possesses knowledge that other nodes do not have. This attribute greatly increases the value of the node. The calculation equation is:

$$\text{Uniqueness}(\text{data}, S_i) = \frac{P(\text{data}, S_i) - BP(S_i)}{P(S_i)}$$

Based on the uniqueness model, we propose a node data contribution model that refers to the degree to which a node provides knowledge relative to the entire decentralized network:

$$\text{Contribution}(\text{data}) = \sum_{S_i \in \text{Data}} \text{SF} \cdot \text{Uniqueness}(\text{data}, S_i)$$

where SF $\in [0, 1]$, SF > 0. In Equation (9), SC represents the query frequency of entity S_i, which also represents the impact of a transaction on data quality —one aspect of user behavior. User feedback represents user assessment of the availability, correctness, and other dimensions of the entity, which greatly affects the degree of node data contribution. The smaller the item is, the more reliable the data are, and k is the user feedback adjustment coefficient.

To compute and update these dimensions, we present an RDF data set entity record table to record uniqueness coefficients, query frequency, user feedback, and other entity information. Its structure is shown in Figure 2 [Figure 2: see original paper].

Figure 2. RDF data set entities record table structure

The uniqueness coefficient represents the degree of uniqueness of an entity in the entire knowledge graph. Query frequency refers to the number of times the entity appears in query results, reflecting the impact of transaction information on quality assessment. User feedback refers to the user' s comprehensive assessment of the entity' s availability, correctness, and other dimensions, with a default value of 0. Query frequency and user feedback embody the key role of user behavior in quality assessment. Contribution(data) can be calculated through these log indicators.

For example, nodes A and B contain the triples shown in Table 4 , demonstrating how uniqueness influences nodes. First, we calculate the uniqueness coefficient of each subject and generate an RDF data set entity record table (with default query frequency of 1 and default user feedback of 0). Then we receive user queries and feedback to update the RDF data set entity record table. Finally, we introduce the calculation of the uniqueness model.

Table 4. Entities in node A and node B

Node ID	Triple
A	<s1,p1,o1>
A	<s1,p3,o3>
A	<s3,p2,o2>
B	<s1,p2,o2>
B	<s2,p1,o2>
B	<s1,p3,o5>

For subject $s1$ in node A, a calculation example of entity uniqueness is given here:

$$\text{Uniqueness} = \frac{P(\text{data}, s1) - BP(s1)}{P(s1)} = \frac{2 - 0.5}{2} = 75\%$$

The received queries and feedback: - Query statement: `select ?S where ?S ?P o2` - Query results: <s1,p2,o2>, <s2,p1,o2>, <s3,p2,o2> - User feedback: $s3$ results are unreliable

We update the RDF data set entity record table to obtain the results shown in Figure 3 Figure 3: see original paper.

Figure 3. RDF data set entities record the table transformation process

The influence of uniqueness, transaction information, and user feedback on the whole is calculated as follows:

$$\text{Uniqueness}(\text{data}) = 1.0 \cdot 2 = 0 \quad (\text{when } k = 2)$$

The effect of user behavior is increased by adjusting the value of k in Equation (9). Taking $k = 2$ to calculate the overall impact model of uniqueness on node A: Contribution(data) = 3.5.

3.4 User Behavior

In data quality assessment, user behavior is indispensable, and only quality assessment combined with user behavior can be meaningful. User behavior can dynamically and comprehensively adjust data quality assessment from the outside, serving as a common method to compensate for model computing limitations under different scenarios and conditions. User behavior affects the accuracy, credibility, and other dimensions of data quality assessment. Furthermore, when combined with blockchain, every transaction belongs to user behavior and becomes an important component of data quality assessment.

For data correctness, this dimension can be influenced by user feedback. Although we have the entire data set, we cannot judge whether a triple is true or

false. User feedback can determine whether the triple is available or not. User feedback on correct and incorrect triples in a query greatly affects the quality of the entire data set.

Transaction information adds value to entities in query results. Assuming node A has entity $s1$, which is not found in other nodes, and the entity query frequency is high, the node will contribute more to the entire knowledge graph. Similarly, if node A and other nodes have the same entity $s2$ with identical entity attributes, then entity $s2$ in node A contributes less to the knowledge graph. In this paper, the query distribution in a query is combined with subject relevance, uniqueness, and the final overall impact of data quality assessment results.

User behavior consists of two parts in this system: user query (i.e., a transaction) and user feedback. Both parts affect quality assessment results. Unlike initial static calculation of node quality assessment, user behavior impact is dynamic because user behavior triggers much faster than quality updates. Therefore, this paper proposes query logs and RDF entity record tables to cache the results of user actions.

3.5 Node Data Quality

Node data quality comprises several components, including dimensions from the RDF inspection report, data completeness, verifiability, user feedback, etc. The comprehensive model of data quality assessment is given here:

$$\text{Quality(data)} = k_1 \cdot \text{QRDF(data)} + k_2 \cdot \text{Contribution(data)} + k_3 \cdot \text{Verifiability(data)}$$

where $k_1 > 0$, $k_2 > 0$, $k_3 > 0$. In Equation (12), QRDF(data) represents the inspection report of Data; Contribution(data) represents the degree of contribution of the data set Data; Verifiability(data) represents the validity of the data set Data. The second item in the equation accounts for a large proportion because if the unique knowledge possessed by a node is widely demanded by users, it demonstrates the indispensability of that knowledge and indicates higher data quality for the node.

4. DCQA System Design

DCQA is an alliance information interaction system based on the P2P protocol, designed for RDF data storage and multi-node communication. An important feature of P2P is its ability to change the current state of the Internet centered on large websites, returning to "decentralization" and restoring power to users. Currently, its greatest advantage is improved network utilization for users because multiple nodes are interconnected, maximizing user network bandwidth. Alliance information interaction means each node possesses independent knowledge and interacts with other nodes while protecting that knowledge. Users can access from any node with consistent access effects. The system has no central

node and belongs to a decentralized network structure. The system network structure is shown in Figure 4 Figure 4: see original paper. The logical node structure is shown in Figure 4(B). Nodes are logically divided into three layers: service layer, protocol layer, and data layer. The service layer primarily provides node Web services, including query interface, node basic information interface, node quality information interface, etc. The protocol layer includes the HTTP protocol and encapsulated P2P protocol. The data layer includes RDF data sets, basic node information, query logs, quality assessment ledgers, and RDF entity record tables. Basic node information refers to fundamental information about nodes in the entire network, including routing information and quality assessment information. The quality assessment ledger is a concrete implementation of the quality assessment blockchain, recording transaction information and quality assessment updates. The query log supplements transaction information combined with knowledge quality assessment.

Figure 4. DCQA system network structure

4.1 Node Connectivity Calculation

In decentralized systems, each node possesses its own knowledge, and knowledge independence within nodes is protected. To calculate uniqueness and other dimensions in each node's data quality assessment, information must be communicated between nodes. Therefore, we propose a node-communicative calculation method.

When connection calculation is necessary, the initiator establishes a temporary node to join the decentralized system. Temporary nodes have independent flags indicating their temporary status. These nodes provide services that can verify their identity. Figure 5 [Figure 5: see original paper] shows the verification method for temporary nodes.

Figure 5. Temporary node verification process

To ensure the temporary node is indeed created by a specific node in the system, the parent node sends the RDF medical report to the temporary node upon creation. Each node compares the hash value of the parent node' s RDF medical report provided by the parent node and the temporary node. Successful comparison indicates the temporary node was created legally. Simultaneously, to ensure knowledge independence and protection for each node in the decentralized system, nodes do not transfer complete triplet information to the temporary node but rather a combination of `<subject-predicate>`. Finally, after the temporary node calculates the task result, it returns the result to each node in the system, adds the result to the quality assessment calculation, and the temporary node self-destructs.

Figure 6 [Figure 6: see original paper] illustrates the process of ensuring temporary node destruction. For security, after a temporary node' s destruction, two things must be ensured: (1) the temporary node no longer exists; and (2) the

parent node A that created the temporary node does not steal information from other nodes. For the first requirement, each node accesses the destructor information of a temporary node and updates routing information if it is unreachable. If the temporary node no longer exists, each node makes its own query for node A' s RDF data, and node A provides a service that only calculates the hash of the query result without returning the result. When the temporary node is destroyed, node A is queried to obtain the hash value, which is checked and verified against the hash when the temporary node was created.

Current and late comparisons indicate that the node content has not changed. If comparison results are inconsistent, the node may have illegally stolen other nodes' information. If hash results are inconsistent, node A will be removed from the decentralized network.

Figure 6. Temporary node destruction process

4.2 Advantages of Using Blockchain to Record Quality Assessment Results

This paper uses blockchain to record transaction information and quality assessment result information. The advantages of using blockchain for quality assessment recording are as follows:

- **Structural security**: Quality assessment certificates are issued without an authority center. Nodes mutually authenticate, and each node maintains backups of other nodes' quality assessments. This prevents quality assessment results from becoming non-credible due to central node collapse or attacks.
- **Information security**: Prevents nodes from tampering with quality assessment results. The consensus mechanism prevents Byzantine attacks. The entire network maintains a super-ledger of quality assessment results. If a malicious node attempts to forge quality information, it can be effectively screened by the entire network. Malicious nodes would need to control more than 50 percent of nodes to launch forgery attacks, which carries a high cost.
- **Quality result update mechanism**: Previously, RDF data required recertification and republication after each update. Using blockchain recording allows updates within the node itself, with update logs written to the block and recorded in the quality assessment ledger.
- **Update log traceability**: Each update record has a basis, enabling quality assessment review from the first generation to the final change.

With these advantages, the DCQA system structure is robust. Nodes in the DCQA system mutually authenticate, and quality assessment results are backed up across nodes. There are no central nodes in DCQA, effectively preventing credibility issues when central nodes collapse or are attacked. Simultaneously,

the model uses a consensus mechanism to effectively prevent malicious nodes from tampering with quality assessment results, ensuring information security. For each RDF data update at a node, the DCQA model records the quality assessment result and update log into the quality assessment ledger, effectively implementing a dynamic update mechanism for quality assessment results with reference to corresponding update records. Such quality assessment results not only enable users to use RDF data sets with confidence but also guide the system to provide users with more cost-effective query results.

4.3 Quality Assessment Blockchain Construction Method

This paper also discusses the application of incremental builds because nodes are dynamically added to the system. When a new node joins the system, node quality assessment is performed and results are synchronized to each node' s quality assessment ledger. This section mainly introduces the specific process of generating and synchronizing quality assessment results when a new node joins the decentralized system.

First, when a new node joins the decentralized system, quality assessment is required through the following process:

1. The new node enters the decentralized system, synchronizing the quality assessment account.
2. Create a new temporary node and notify other nodes.
3. Provide the RDF data inspection report to the temporary node.
4. Each node in the network records the RDF inspection report hash value of the new node for later comparison.
5. The temporary node calculates the hash value of the parent node' s inspection report. Each node compares this value with the hash value provided by the parent node to verify the temporary node' s validity.
6. Each node provides a combination of subject-predicate pairs.
7. The temporary node computes completeness, relevance, uniqueness, and other related attributes for each node and returns them to each node.
8. Each node generates an RDF entity record table.
9. The quality assessment calculation result is generated.
10. Quality assessment results are broadcast to other nodes for signature.
11. Write the initial quality assessment hash record into the Merkle tree.
12. The temporary node self-destructs.
13. Each node verifies that the temporary node is fully destroyed and then updates the route.

According to the above process, the initial quality assessment of the new node in the decentralized system is completed. The quality assessment result becomes the first item in the quality assessment result update logs, and subsequent quality assessment result updates start from this record.

Figure 7 [Figure 7: see original paper] shows the process of node quality assessment record generation. The node-generated quality assessment result is

broadcast to other nodes for signature. The signature here is a chain signature —that is, according to the order of joining the system, each node signs, and finally returns to the temporary node. After obtaining all node signatures, the first quality assessment record is generated.

Figure 7. Node quality assessment record generation process

Blockchain storage and recording of transaction records can form a ledger that prevents transaction information from being forged or modified. Similarly, storing quality assessment results in the same way ensures each update has the same effect, with each update based on the previous quality assessment results. The update log contains block ID, update items, signature, and timestamp.

An update item is a dimension that is reduced or promoted in one update. Recording update items occupies more space than recording only incremental results. However, if only incremental quality assessment results were recorded, quality assessment result updates could not be fully represented. Therefore, results are updated, and each block calculates a new quality assessment result by itself. Signature refers to the signature information of each block, indicating that each block knows the record exists, and the signature item is a list. Normally, the signature list should include signatures of all blocks. The block ID describes a block that updates the quality assessment result.

We can form a time tree according to the timestamp. Each update is based on which items of the last quality assessment result are updated, realizing the relationship between updates, which can effectively prevent quality assessment update fraud.

After the update log is generated, it is released to each block for signature as shown in Figure 8 [Figure 8: see original paper]. After all nodes have signed, the results are recorded. Here we use the Merkle tree [**?**] to store quality assessment results. A Merkle tree is a tree where every leaf node is labeled with the hash of a data block and every non-leaf node is labeled with the cryptographic hash of its child node labels. Hash trees allow efficient and secure verification of large data structure contents. Using the Merkle tree can effectively prevent result tampering—any tampering will cause a change to the hash value stored by the root node. When the hash value is inconsistent with other nodes, the quality assessment result in the block is considered unreliable, and the node will be removed from the decentralized network.

Figure 8. Account schematic diagram of quality assessment

4.4 Interaction between User Behavior and Quality Assessment

There is a bidirectional relationship between user behavior and quality assessment. User behavior in this paper includes two parts: transaction conclusion and user feedback. Each query is essentially a transaction. Here we take query as an example to explain how query and quality assessment interact.

According to Equation (9), query affects quality assessment. The higher the query frequency, the higher the quality assessment. The following is the processing flow of a query:

1. Receive the query statement and execute the query.
2. Feed back the results to users.
3. Update the query log.
4. Update the RDF data set entity record table.
5. Generate transaction information.
6. Wait for T time for user feedback.
7. Update the data entity record table and query log if user feedback is available.
8. End this query; subsequent operations will not affect this query.

From the above process, we can conclude that maintaining the query log and the RDF data set entity record table does not occupy query response time but is performed after query results are fed back to the user. This process reflects how queries and user feedback affect quality assessment. Likewise, quality assessment guides the query mechanism to return more cost-effective results. The following are the basic rules of the system:

- When the cost of query results is the same, data from nodes with high service quality are preferred.
- When query results differ, data from nodes with a higher ratio of result data to price are recommended.
- When part of the query results are the same but some nodes have unique entities, higher-quality node data of query results are recommended.

5. Experiment Analysis

The experimental objectives are to verify model verifiability, data integrity, the impact of user feedback on quality assessment results, and whether the quality assessment ledger is secure. This section proposes a decentralized system based on the model and system design from sections 3 and 4, and conducts a series of experiments.

Experimental environment: MacBook Pro laptop with a 2.6 GHz Intel Core i7 processor and 16GB 2133 MHz LPDDR3 memory. In this environment, we simulated a decentralized system with 6 nodes. The protocols used include the HTTP protocol and the P2P protocol.

The experiment uses the Archives Hub data set, a sample data set of descriptions of archive collections held on the Archives Hub (a UK aggregator) and output as linked data. The Hub linked data provides perspective on the people, organizations, subjects, and places connected with the described archives. The data set file size is 71.8M, with 106,919 entities, 51,411 unique subjects, 141 unique predicates, 104,408 unique objects, and 431,088 triples. To highlight the importance of quality assessment dimensions such as verifiability, completeness,

and uniqueness in the model, we divided the Archives Hub data set into six parts labeled AH1 to AH6. Table 5 shows basic information for the six data sets. To increase completeness judgment, we selectively replicated some node information across AH1 to AH6.

Table 5. Basic information of the experimental data set

Data set	File size (M)	The number of entities	The number of triples	The number of unique subject and predicate	RDF inspection report
AH1	16,748	20,514	28,361	6,945/89	0.75
AH2	20,876	28,366	39,705	9,271/46	0.68
AH3	20,514	40,803	56,722	2,676/43	0.81
AH4	28,366	43,590	79,410	4,576/66	0.72
AH5	40,803	102,099	124,796	12,983/29	0.85
AH6	43,590	106,919	102,099	14,970/101	0.77

The RDF inspection report in Table 5 is calculated according to Equation (3), where $k_1 = 1$, $k_2 = 1$, $k_3 = 1$, and $k_4 = 1$. All coefficient values are set to 1 because in this system we assume equal weights for the four factors affecting RDF data quality mentioned previously.

Before the system accepts queries, the system does not activate verifiability and uniqueness in the model (Equation (9)), so the RDF inspection report result serves as the initial quality assessment result of the node.

5.1.1 Verifiable Model Validation

First, we verify the verifiability of the model using statement 1 and statement 2 to activate the model. The activation process involves making 100 queries after the decentralized system is set up. The language used is SPARQL [**?**], a query language and data acquisition protocol developed for RDF. We calculate and update the new quality assessment results for each node.

Statement 1: `select * { ?s ?p ?o } where { <http://api.talis.com/stores/locah/items/13052833 ?p ?o . }`
Hit result: AH1, AH2, AH6

Statement 2: `select * { ?s ?p ?o } where { <http://data.archiveshub.ac.uk/id/person/aacr2/ma ?p ?o . }`
Hit result: AH6

Once quality assessment results are updated, we update the entities hit by statement 1 in the data set after the 100th query and then execute statement 1 again. The query result record is shown in Figure 9 [Figure 9: see original paper].

Figure 9. Impact of verifiability on quality assessment

According to the records in Figure 9, the quality of AH1 and AH6 remains stable for the first 100 queries because the data set itself has not changed, and query result consistency is stable. At the 101st query, verifiability landslides occur, causing a sharp decline in quality assessment results because data in AH1 and AH6 were updated, making query results inconsistent with previous ones. For users, frequent data changes are undesirable, so a large-scale decline in quality assessment meets expectations. Simultaneously, as the number of queries increases, quality assessment steadily increases linearly. If data updates occur during this process, data quality will still degrade. Comparing quality assessment results between AH1 and AH6, we find that AH6' s quality degradation is lower than AH1' s after 100 queries because statement 2 was executed during the activation process, resulting in the AH6 query log containing two parts, and the updated data do not affect statement 2 hit results. Therefore, AH6' s quality decline is less. By comparing AH6 with AH1, we verify that the verifiability model' s granularity is at the log level.

5.1.2 Impact of Data Completeness and User Feedback on Quality Assessment

Table 6 shows an RDF entity record generated after model activation. As the number of queries increases, node data quality gradually improves.

Table 6. Records for RDF entities in AH1 and AH6

Data set	Entity	Uniqueness coefficient	Query frequency	User feedback
AH1	self	0.3	100	0
AH6	self	0.3	100	0
AH6	Martindorothyfreeborn	0.85	100	0

Note: self stands for entity at http://api.talis.com/stores/locah/items/1305283343810#self. Martindorothyfreeborn stands for entity at http://data.archiveshub.ac.uk/id/person/aacr2/martindorothyfreebor

Since the entity self exists in other data sets, the uniqueness coefficients of AH1 and AH6 are lower. We executed statements 1 and 2 multiple times and randomly added user feedback after 500 executions. Figure 10 [Figure 10: see original paper] shows quality changes in AH1 and AH6 during this process.

Figure 10. AH1 and AH6 verification uniqueness model

In Figure 10, the quality of AH1 and AH6 has steadily increased due to growing query frequency. AH6 grew more rapidly because it involves multiple subjects. After 500 queries, less than 50 percent of users added feedback interference, slowing quality increase. After 700 queries, interference was adjusted to more than 50 percent, reducing quality. We observe that quality is related to the user feedback coefficient, which can be used to adjust the impact of user behavior. We also find that AH6' s increases and decreases are significantly higher than

AH1' s because: (1) AH6 involves more subjects; and (2) AH6 has a subject with a higher uniqueness coefficient.

5.2 Quality Assessment Ledger Safety

Although blockchain solves security problems, 51% attacks can still occur. That is, if someone controls more than 51% of the entire network' s computing power, they can preemptively create a longer chain of forged transactions in a race attack. This paper calculates the difficulty of controlling each node and analyzes which nodes will launch attacks from an interest perspective.

If nodes A, B, C, and D contain subject $s1$, with attribute counts of 9, 5, 31, and 7 respectively, and the common attribute count is 5, then for each query, the initial contributions of A, B, C, and D are 0.105, 0, 0.763, and 0.05 respectively. Assuming the prices of A, B, C, and D are the same, the cost for node B to launch an attack is:

$$\text{Cost} = \sum_{\text{node}i} \text{Uniqueness}(\text{data}, S_j) \cdot \text{price}(\text{node}_i)$$

where node_i is the number of nodes intended to be bribed. The relationship between the number of nodes and cost value is shown in Figure 11 [Figure 11: see original paper]. We can see that the more nodes are bribed and the higher the data quality of nodes intended to be bribed, the higher the cost. To gain 51% support, the cost is much greater than the benefit. As the network grows, the probability of launching a 51% attack approaches 0.

Figure 11. Relationship between node number and cost value

6. Conclusion

This paper proposes a method for assessing and updating RDF data quality in the context of the rapidly developing Web and decentralized systems. First, it indicates the significance of this research and the importance of achieving RDF data quality assessment in decentralized systems. Then, an RDF data quality assessment model for decentralized systems is proposed. Finally, based on the node quality assessment model, a decentralized quality assessment system DCQA is designed and implemented. We also proposed and discussed in detail the use of blockchain to store quality assessment results. The obtained RDF data quality assessment results combine blockchain advantages without requiring quality assessment certificates from authoritative centers, effectively preventing tampering and supporting dynamic updates of record queries. The purpose of this paper is to promote research and development of RDF data quality assessment in decentralized systems and provide users with better services. In the future, we will carry out research in the following aspects: (1) new dimensions of quality assessment in decentralized environments, and (2) the impact of price factors on the model.

Author Contributions

All authors made significant contributions to this work. L. Huang (huangli82@wust.edu.cn) is the leader of this work. She designed the RDF data quality assessment mechanism and DCQA system framework in a decentralized system. Z. Liu (540900559@qq.com) and F. Xu (xuff@wust.edu.cn) summarized the methodology part of this paper. J. Gu (simon@wust.edu.cn), as the corresponding author, summarized and drafted this paper. All authors made meaningful contributions to the revision of the paper.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No: U1836118, Grant No: 61602350), the Key Projects of National Social Science Foundation of China (Grant No: 11&ZD189), and the Scientific Research Project of Education Department of Hubei Province (Grant No: B2019008). We thank graduate students Yansong Wang and Ren Hui for their professional guidance in implementing the DCQA system, as well as Professor Lynda Hardman from the Information Access Research Group, Centrum Wiskunde & Informatica (CWI), Amsterdam, The Netherlands, for her suggestions for modification and improvement of this paper.

References

1. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Available at: https://bitcoin.org/en/bitcoin-paper.
2. M. Swan. Blockchain: Blueprint for a new economy. Sebastopol, CA: O' Reilly Media, 2015.
3. R. Beck, J. S. Czepluch, N. Lollike & S. Malone. Blockchain—The gateway to trust-free cryptographic transactions. In: *The Twenty-Fourth European Conference on Information Systems (ECIS)*, 2016, pp. 1–14. Available at: http://aisel.aisnet.org/ecis2016_{rp}/153.
4. T. Berners-Lee, J. Hendler & O. Lassila. The semantic web. *Scientific American* 284(5) (2001), 28–37. doi: 10.1038/scientificamerican0501-34.
5. S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann & I. Horrocks. The semantic web: The roles of XML and RDF. *IEEE Internet Computing* 4(5)(2000), 63–73. doi: 10.1109/4236.877487.
6. A. Newitz. Web 3.0. *New Scientist* 197(2647)(2008), 42–43. doi: 10.1016/S0262-4079(08)60674-0.
7. C. Bizer, R. Cyganiak & T. Heath. How to publish linked data on the web. Available at: http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/LinkedDataTutorial/.
8. A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres & S. Decker. An empirical survey of linked data conformance. *Journal of Web Semantics* 14(2012), 14–44. doi: 10.1016/j.websem.2012.02.001.
9. J. Kandari. Information quality on the World Wide Web: A user perspec-

tive. PhD dissertation, University of Nebraska-Lincoln, 2010. Available at: doi: 10.1504/IJIQ.2011.043784.
10. A. Assaf & A. Senart. Data quality principles in the semantic web. In: *IEEE Sixth International Conference on Semantic Computing*, 2012, pp. 226–229. doi: 10.1109/ICSC.2012.39.
11. A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann & S. Aue. Quality assessment for linked data: A survey. *Semantic Web* 7(1)(2016), 63–93. Available at: http://www.semantic-web-journal.net/system/files/swj773.pdf.
12. C. Bizer & R. Cyganiak. Quality-driven information filtering using the WIQA policy framework. *Journal of Web Semantics*, 7(1)(2009), 1–10. doi: 10.2139/ssrn.3199414.
13. D. Magazzeni, P. McBurney & W. Nash. Validation and verification of smart contracts: A research agenda. *Computer* 50(9)(2017), 50–57. doi: 10.1109/MC.2017.3571045.
14. R. C. Merkle. Protocols for public key cryptosystems. In: *1980 IEEE Symposium on Security and Privacy*. IEEE, 1980: 122–122. doi: 10.1109/SP.1980.10006.
15. SPARQL. Available at: https://www.w3.org/TR/sparql11-query/.

Author Biography

Li Huang is currently an associate professor of computer science in the College of Computer Science and Technology, Wuhan University of Science and Technology. She received her PhD in computer science from Huazhong University of Science and Technology in 2011. Her research interests include data management, semantic Web, and knowledge. ORCID: 0000-0001-5702-2321

Zhenzhen Liu is currently a postgraduate student in the College of Computer Science and Technology, Wuhan University of Science and Technology. He received a bachelor' s degree from Wuhan University of Science and Technology in 2017. His research interests include semantic Web, knowledge graph, and linked data quality assessment. ORCID: 0000-0003-2066-312X

Fangfang Xu received her B.S. degree from the College of Computer Science and Technology at Wuhan University of Science and Technology (WUST) in 2012 and the M.S. degree from College of Computer Science and Technology at WUST in 2015. Now she is an engineer at the College of Computer Science and Technology, WUST. Her research interest is semantic computing. ORCID: 0000-0003-1057-6444

Jinguang Gu is currently a professor of the College of Computer Science and Technology, Wuhan University of Science and Technology (WUST) and vice dean of Institute of Big Data Science and Engineering, WUST. He received his PhD in software theory from Computer School of Wuhan University in 2005. His research interests include knowledge graph, semantic Web, and distributed and service computing. ORCID: 0000-0002-8823-8480