OPEN ACCESS
Vietnam Journal of Computer Science
Vol. 8, No. 4 (2021) 471–491

DOI: 10.1142/S2196888821500202

Validating Knowledge Contents with Blockchain-Assisted Gamified Crowdsourcing

Helun Bu*

Graduate School of Information Science and Engineering
Ritsumeikan University
1-1-1 Noji Higashi, Kusatsu, Shiga 525-8577 Japan
**is0385pr@ed.ritsumei.ac.jp*

Kazuhiro Kuwabara

College of Information Science and Engineering
Ritsumeikan University
1-1-1 Noji Higashi, Kusatsu, Shiga 525-8577 Japan

Received 15 June 2020
Accepted 24 November 2020
Published 19 January 2021

This paper presents the use of gamified crowdsourcing for knowledge content validation. Constructing a high-quality knowledge base is crucial for building an intelligent system. We develop a refinement process for the knowledge base of our word retrieval assistant system, where each piece of knowledge is represented as a triple. To validate triples acquired from various sources, we introduce yes/no quizzes and present them to many casual users for their inputs. Only the triples voted "yes" by a sufficient number of users are incorporated into the main knowledge base. Users are incentivized by rewards based on their contribution to the validation process. To ensure transparency of the reward-giving process, blockchain is utilized to store logs of the users' inputs from which the rewards are calculated. Different strategies are also proposed for selecting the next quiz. The simulation results indicate that the proposed approach has the potential to validate knowledge contents. This paper is a revised version of our conference paper presented at the 12th Asian Conference on Intelligent Information and Database Systems (ACIIDS 2020).

Keywords: Blockchain; knowledge refinement; gamified crowdsourcing.

1. Introduction

This paper describes an application of blockchain-assisted gamified crowdsourcing for knowledge refinement, focusing on a knowledge content validation process.

*Corresponding author.

Constructing a high-quality knowledge base is important for developing an intelligent system, and knowledge refinement is necessary to increase the value of the knowledge base. Several approaches to refining knowledge represented as knowledge graphs have been reported.[1] One promising approach is to harness the power of many users, for example, through crowdsourcing.[2] However, the inherent problem in crowdsourcing is maintaining a high-quality output. It is important to motivate users (or workers) to produce robust results.

Gamification is one way to keep users motivated in crowdsourcing.[3] Gamification is the process of introducing game-like elements to a non-gaming context. Games with a purpose (GWAP) is an example of gamification, where intended tasks are executed as by-products of playing games.[4] GWAP is also applied to refine knowledge graphs.[5] In the ESP game, which is a notable example of GWAP, two users who do not communicate with each other are asked to label an image. Points are given to a user who puts the same label as their paired user.[6] By devising proper game rules, an incentive can be given to users to input correct answers.

Point systems are expected to motivate users to earn more points and complete more tasks with correct results. In such situations, it is important to correctly calculate points. If points cannot be calculated at the time users are playing the game or completing the task, the points will be calculated later according to the user's past contribution; thus, it is necessary to ensure that the users' records or input logs that are used for the points calculation are not altered in any way. The point calculation process must be transparent so that any user can examine the basis for the point calculation. If we assume that the game is operated by a single trusted entity, as long as the operator is trusted, point calculations can also be trusted. In that case, a distributed database is an effective way to store data in a distributed way, while data integrity is maintained. However, if we assume a trust-less environment where no single trusted entity exists, we need a mechanism to keep the transparency and tamper-resistance nature of the data. To tackle this problem, we utilize blockchain, which is a distributed ledger technology that has been proposed as the basis of cryptocurrency.[7] Blockchain allows the data to be stored and shared over the network, and it is guaranteed to be free from tampering.

In this paper, we focus on the example of the knowledge base used in our word retrieval assistant system,[8] which is intended to support people with word-finding difficulties. Through a series of questions and answers, the system attempts to guess what the person wants to express but cannot find a name for. The knowledge base is needed to formulate questions to ask. For this system, we accumulate knowledge contents from various sources such as scraping websites or obtaining inputs from human users. Because of the nature of this system, its knowledge contents should cover topics that often appear in everyday conversation, such as things related to daily life and not in some specialized domains. Thus, the participation of many casual users to construct the knowledge base would be effective.

However, when many casual users contribute to knowledge contents, the quality of the knowledge may be compromised. To ensure high-quality knowledge, we first store newly acquired pieces of knowledge into a temporary knowledge base and move only the validated knowledge into the main knowledge base.

In this validation process, we employ a concept of gamified crowdsourcing similar to the one explored by Kurita *et al.*[9] More specifically, we make simple yes/no quizzes from the contents of the temporary knowledge base, which we present to the users. When enough votes for agreeing with contents are accumulated, the corresponding knowledge contents are judged to be correct, and are incorporated into the main knowledge base. As an incentive to users, we award points based on a user's past inputs. If they contribute to the validation process, they are given rewards. Blockchain technology is utilized to record the users' input and ensure the transparency of the reward calculation.

This paper is a revised version of our paper presented at the 12th Asian Conference on Intelligent Information and Database Systems (ACIIDS 2020).[10] There are two major differences from the original paper: (1) we clarify the organization of a blockchain-assisted gamified crowdsourcing system and implement a prototype system and (2) using the prototype system we developed, we conduct simulation experiments with additional parameters to investigate the characteristics of the proposed approach including different task selection strategies. We also conduct a small real-life experiment and confirm the effectiveness of the proposed approach.

The remainder of the paper is organized as follows. Section 2 describes some related work focusing on the application of blockchain for handling knowledge, and Section 3 presents the knowledge refinement process. Section 4 describes the organization of the prototype implementation we used for experiments, and in Sec. 5, we report the simulation experiments that use the prototype and discuss their results. Section 6 presents a small real-life experiment we conducted to verify the knowledge refinement process with human users, and Sec. 7 concludes this paper.

2. Related Work

Blockchain was developed as an underlying technology for cryptocurrency and was intended for use as the public ledger for transactions on a network. Blockchain is now applied not only to cryptocurrency but also to other areas, such as health care, data provenance, and mobile communication networks.[11] Blockchain is also used as a platform for crowdsourcing. For example, CrowdBC is a blockchain-based decentralized crowdsourcing platform that makes use of smart contracts to achieve reliability with low service cost.[12] zkCrowd is another blockchain-based platform that separates a blockchain into a public chain and private subchains so that users' privacy can be maintained while keeping transaction transparency.[13] These systems are intended to provide a domain-independent platform for crowdsourcing. Furthermore, blockchain is used in the game industry as a basis for maintaining rule transparency.[14] Applications to gamified crowdsourcing are also mentioned as potential areas to be explored.

Another notable application area is the Internet of Things (IoT). The mobile crowd sensing (MCS) platform was developed using blockchain.[15] Here, a market-like idea was introduced in handling sensed data to maintain both privacy and security. The proposed platform focuses particularly on an incentive mechanism for rewards. A blockchain is also utilized to construct a *knowledge market* in which IoT systems that perform artificial intelligence (AI) tasks at the edge of networks can exchange knowledge in a peer-to-peer fashion.[16] For AI systems that require a large dataset, it is important to provide a proper incentive mechanism to continuously update the dataset.[17] Blockchain is also used as a basis for such a platform where a concept of gamification is introduced.

For knowledge management in enterprises, *Knowledge Blockchain* was proposed[18] to audit knowledge evolution and provide proof of provenance of knowledge. Blockchain is also utilized in access control of a knowledge management system.[19] A platform for smart city was also developed using blockchain.[20] This platform called WeValue is intended to be a platform for social value exchange and co-creation. It assumes transactions between unknown people in a trust-less distributed environment.

Blockchain can be viewed as a decentralized database where data from participants are stored transparently. Knowledge graphs represented in the Resource Description Framework (RDF) are stored using a blockchain technology, called GraphChain.[21] Blockchain is also applied to the decentralized construction of knowledge graphs.[22] In this system, company-level domain knowledge about employees' skills is constructed from the participation of employees in the company. This system also introduces a voting scheme and a reward mechanism for employees who contribute to knowledge construction. AUDABLOK is another example of a blockchain application; it was proposed as a software framework to allow citizens to participate in refining open data.[23] In AUDABLOK, blockchain is utilized to audit users' contributions and provide rewards to users.

In this paper, we emphasize on the validation process of knowledge contents by casual users, where no centralized trusted authority is assumed. We also conduct simulation experiments to examine its characteristics.

3. Knowledge Refinement Process

3.1. *Target knowledge base*

In this study, we chose the knowledge contents used in our word retrieval assistant system[8] as a target knowledge base. A typical problem for people with aphasia is word-finding difficulty, where they have a clear image of what they want to say, but cannot recall a proper word to express it. This is similar to the situation when you visit a foreign country and you do not know how to say what you want in the local language.

For this kind of difficulty, a human caregiver called a conversation partner often provides assistance by asking a series of questions, such as *Is it food?* or *Is its color* *red?* Through this kind of question and answer, the conversation partner extracts the name of the thing the person with aphasia wants to express.

The word retrieval assistant system is intended to provide a similar function to a human conversation partner, but using a computer. The system contains a knowledge base about relevant things and generates appropriate questions to ask the user. According to the user's replies, the next question to ask is determined.

In the early prototype system, knowledge is represented as a triple of *subject*, *predicate*, and *object* as in the RDF. For example, the fact that *the color of the apple is red* can be represented as `(<apple>, <color>, <red>)`.

3.2. *Knowledge acquisition and refinement*

It is important to acquire enough knowledge for the word retrieval assistant system to work properly. There are several methods to construct a knowledge base. System developers construct knowledge contents manually from scratch, or the system lets a user input the correct word when it is not produced by the system.[24] We may extract knowledge from data available on the Internet, such as Wikipedia. Alternatively, we may use a framework of crowdsourcing to elicit inputs from many casual users. When data for a triple are needed, we may present an input form that consists of three items, one or two of which are blank so that a user can fill them in. To facilitate such a process, gamification concepts can be applied. For example, such an input form can be presented as a quiz to a user; when a user answers the quiz, points are given to the user.

One of the problems with data acquired using such methods is that they may not be correct. To assure the knowledge contents are of high quality, it is necessary to validate the correctness of newly acquired triples before they are used. The acquired knowledge contents are first stored in the temporary knowledge base, and the knowledge contents validated by users are moved into the main knowledge base.

To validate the contents of the temporary knowledge base, we employ a yes/no quiz. Suppose that the triple `(<apple>, <color>, <red>)` is in the temporary knowledge base. A yes/no quiz is presented to a user to ask whether the color of the apple is red. The user may answer YES, NO, or DON'T KNOW. If a certain number of YES votes are obtained, the triple is considered to be correct and is moved into the main knowledge base.

This process is formulated into a kind of game. After a user starts a game, a yes/no quiz sentence is presented with the possible choices of YES, NO, DON'T KNOW, and END. If the user answers END, the game session ends.

For triples that are used in the game session, we check whether enough votes (YES or NO) are obtained for the triple as the results of the game playing so far. We set a certain threshold S, and check whether a triple has accumulated more than S votes. If the triple has accumulated more than S votes and the ratio of YES votes over the total number of votes for the triple is greater than a threshold α, then the triple is judged to be correct. The triple is removed from the temporary knowledge base and moved into the main knowledge base. Similarly, if the ratio of NO votes over the total number of votes for the triple is greater than a threshold β, the triple is judged to be incorrect and is removed from the temporal knowledge base.

When a triple is judged to be correct, the users who voted YES for this triple are given rewards. Similarly, when a triple is judged to be incorrect, the users who voted NO for this triple are given rewards. This check is performed for all the triples that appeared in the game session.

3.3. *Use of blockchain*

As explained in Sec. 3.2, the rewards to be given to a user can be calculated only after a certain number of votes have been collected. The rewards are calculated on the basis of the logs of the game sessions. To ensure transparency of the reward calculation, the game logs are stored in the blockchain. As shown in Fig. 1, when a game session ends, the inputs made by a user are stored in a new block, which includes the user's Id, the Id of the triple corresponding to the quiz the user answered, an answer from the user, and its timestamp. After the new block is added to the blockchain, a history of all users' inputs for the triples in the game session are obtained from the blockchain. If a triple has gathered enough votes to be judged correct or incorrect, its status is updated accordingly, and the rewards for the users who contributed to the votes are determined.

Rewards are calculated from the users' contributions to the validation of triples. Since the triple validation process depends on past results of the game executions, it is necessary to access the game execution logs stored in the blockchain to calculate the rewards. To reduce the data transfer amount from the blockchain server, rewards are calculated at the blockchain server, and the blockchain server is equipped with an application programming interface (API) that returns the reward values. Since the calculation of the rewards basically depends linearly on the size of the log (the length

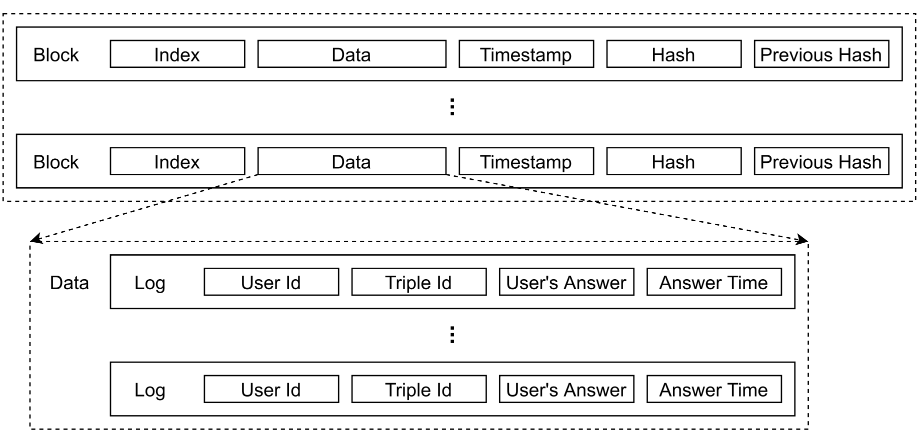


Fig. 1. Data stored in blockchain.

of blockchain), it would not cause a major problem. This is similar in spirit to a smart contract, which defines computations that are triggered when specified conditions are met at the blockchain server.

4. Prototype Implementation

Figure 2 shows the configuration of our early prototype, which is constructed as a chatbot system utilizing LINE messaging service,[a] a popular chat service in Japan and other countries. It is also popular as a platform for the chatbot. A user can access the system via a smartphone so that it can be used in a small amount of spare time.

When the user starts interacting with the chatbot, the chatbot presents a quiz that displays the sentence generated from a target triple and three buttons (YES, NO, DON'T KNOW) to input the user's response along with a button to end the game (Fig. 3(a)). When the game ends, a summary of the game results is presented (Fig. 3(b)).

To implement the user interface with a chatbot that uses the LINE messaging service, the prototype includes a *Game server* that provides a Web API for processing a request to be sent from the LINE server. This request corresponds to the input from a user, and its response should contain the output to present to the user.

The knowledge contents (triples) to be validated are stored in the temporary knowledge base. When a *start* command is sent from a user via the LINE system, a triple to be validated is selected. From the triple, a quiz is created and sent to the user. The response from the user is stored in the *Game server*. When the user ends a game session, the logs from that game session are sent to the *Blockchain server* where they are entered into a new block of the blockchain. The response from the *Blockchain server* includes a list of newly validated or invalidated triples. The newly validated triples are moved from the temporary knowledge base to the main knowledge base, and invalidated triples are removed from the temporary knowledge

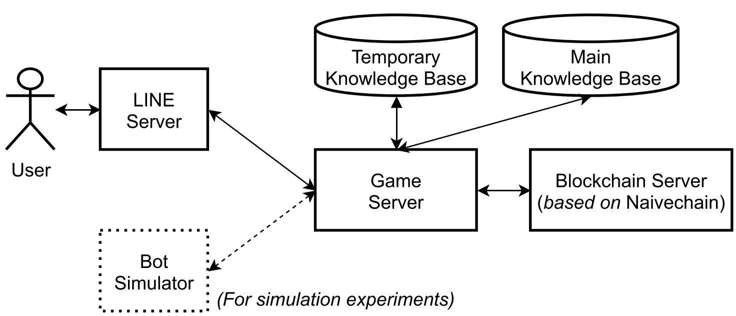
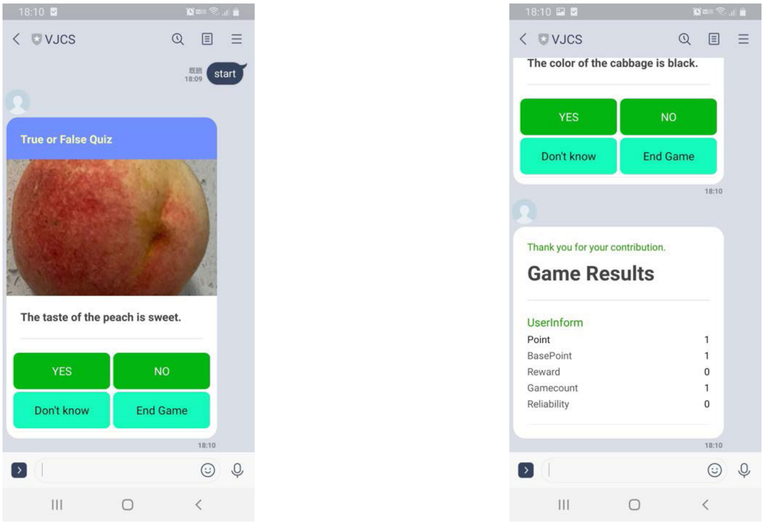


Fig. 2. Prototype configuration.

[a] https://line.me/en/.



(1) During a game session

(2) After a game session ends

Fig. 3. Screenshots of a prototype chat system.

Table 1. Main Web APIs of the blockchain server.

Method	Endpoint	Description
`POST`	`/reset`	Resets a list of the triples to be validated.
`POST`	`/addLog`	Adds a game log to the blockchain, and conducts the reward processing. Returns a list of newly validated/invalidated triples.
`GET`	`/reward`	Returns a reward of a user.

base. Then, the *Game server* presents the points earned in this game session to the user (Fig. 4).

We used the implementation of Naivechain[b] for blockchain. Naivechain implementation includes a Web API to add a new block to the blockchain. We added some Web APIs to interact with the *Game server*, as shown in Table 1.

When the *Blockchain server* receives a game session log, it is stored in a newly created block. In addition, all the triples in the game session logs are checked if they have collected enough votes to be judged correct or incorrect. The results are returned to the *Game server* as described above. In addition, if a triple is newly determined to be correct or not, rewards are given to the users who contributed to making the decision (Fig. 5). The rewards a user earned can be accessed by the Web API of the *Blockchain server*, as mentioned in Sec. 3.3.

[b] https://github.com/lhartikk/naivechain.

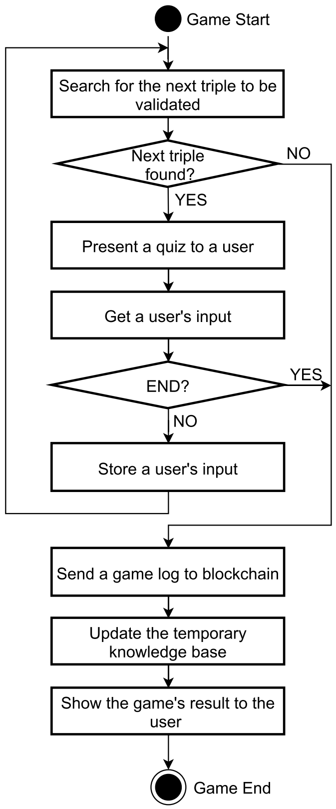


Fig. 4. Flowchart of a game session.

5. Simulation Experiments

5.1. *Purpose and methods*

5.1.1. *Simulation environment*

To examine the characteristics of the proposed approach, we conducted simulation experiments by building a bot program that behaves as a human user (as shown in the dashed lines in Fig. 2). We also added a Web API to the *Game server*, which accepts a request from the bot program. The request from the bot program is similar to a request sent from the LINE server except that it does not contain any authentication information to work with the LINE service.

5.1.2. *User model*

We constructed a simple user model that controls the bot program. This user model contains the following parameters:

- *Accuracy*: user i $(1 \leq i \leq N)$ answers correctly with probability $p_{\mathrm{acc}}(i)$.

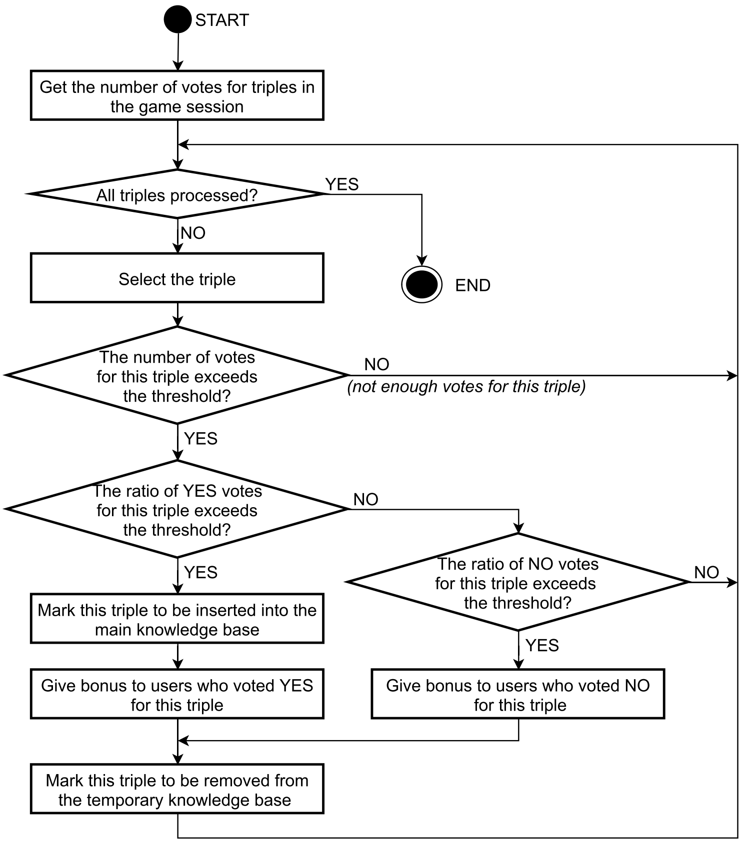


Fig. 5. Reward processing.

- *Confidence*: user i $(1 \leq i \leq N)$ is supposed to answer DON'T KNOW with the probability of $1 - p_{\mathrm{conf}}(i)$.
- *Engagement*: user i $(1 \leq i \leq N)$ is assumed to end a game session after k_i quizzes are answered.

5.1.3. *Simulation conditions*

The conditions of the simulation are set as follows:

- The number of triples (M) is set to 1000, and the number of virtual users (N) is set to 1000.
- All the triples are assumed to be true. Thus, when user i is supposed to answer either YES or NO, user i answers YES with probability $p_{\mathrm{acc}}(i)$.
- A user is selected from a set of N users in sequence, and the selected user plays a game. After k_i quizzes are answered, one game session ends. The value of k_i is set randomly from the range of $[3, 7]$ for all user i.

- When one game session ends, we check whether there are newly validated triples. The threshold for judging a triple to be correct or not (S) is set to 10, and the threshold ratio for judging the triple to be correct (α) is set to 0.8. Thus, a triple that accumulates 10 votes, among which more than eight votes are for YES, is judged to be correct. The users who voted YES for this triple are given 100 points as a reward. After triples to be validated are determined and rewards are calculated, the next user is selected and another game session is started.
- One simulation run is terminated when there are no more triples to present to any user. The same triple will not be used as a game to the same user twice. Once a triple is removed from the temporary knowledge base, it is not used for a game again.

5.2. *Results and discussion*

5.2.1. *Triple selection strategy*

When yes/no quizzes are used to validate triples, the order of the quizzes to present to a user has an impact on efficiency. We tested three variations in selecting the next triple to use for a yes/no quiz. When the next triple was searched for, the triples that had already been used for the target users were removed first. Then, from the remaining triples, we used the following strategies to select the next triple to use as a quiz:

- *Random*: a triple is selected randomly; this is the baseline strategy.
- *Narrow*: a triple that has been presented to users more often is given a higher priority. This method effectively focuses on triples to validate. More specifically, each triple is given a weight equivalent to 10 times the number of times it has been used as a question to users.
- *Wide*: a triple that has been presented to users less often is given a higher priority. This method effectively broadens the range of target triples. More specifically, each triple is given a weight equivalent to 10 times the maximum number of times any triple has been used as a question minus the number of times the particular triple has been used as a question.

In the simulation experiments conducted to examine the effects of different strategies, user i's accuracy ($p_{\mathrm{acc}}(i)$) was randomly set according to the normal distribution with a mean of 0.8 and a standard deviation of 0.067. User i's confidence ($p_{\mathrm{conf}}(i)$) was similarly set according to the normal distribution with a mean of 0.8 and a standard deviation of 0.067.

Simulations were run three times. Table 2 shows the average number of validated triples in these simulation runs. The table indicates that these yes/no quizzes can validate enough triples. The table also shows the average length of the blockchain for each strategy.

Table 2. Average number of validated triples and the length of blockchain for three strategies.

Strategy	*Random*	*Narrow*	*Wide*
Number of validated triples	958	954	954
Length of blockchain	18,053	19,286	19,496

How the number of validated triples changes through a series of game sessions is influenced by a triple selection strategy, as shown in Fig. 6, which plots the number of validated triples against the number of game sessions in one simulation run for three different strategies. As Fig. 6 shows, after a certain number of games, the number of validated triples rapidly increases and then saturates for the *random* and *wide* strategies. Among the three strategies, the *wide* strategy is slow to validate the triples, but the number of validated triples gradually increases compared with the *random* strategy. Although the *narrow* strategy shows a steady increase in the number of validated triples, it is soon surpassed by the other strategies.

5.2.2. *User reliability*

We ran another set of simulations under different conditions, taking user reliability into consideration. Each user's reliability was determined based on their contribution to the validation of triples. The threshold S for determining whether a triple is valid was set according to the sum of the reliability of the users who voted for the triple so

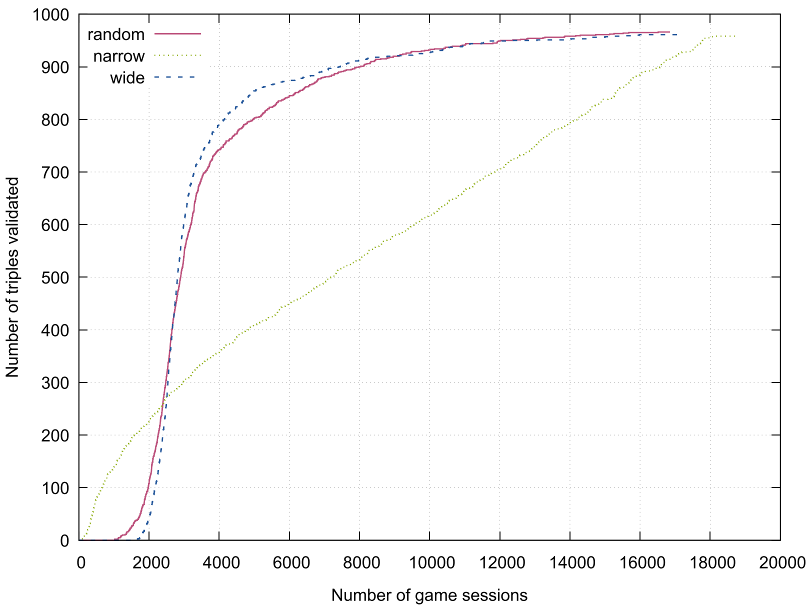


Fig. 6. Changes in the number of triples validated and the number of game sessions in one simulation run.

that fewer users are required to validate a triple when users with a high reliability vote for it. This process was expected to lead to fewer games required for validation.

In the simulation experiments, the reliability of user i, R_i, was set as $R_i = \min(0.2 \times \mathrm{contr}_i + 1, 3)$ where contr_i denotes user i's contribution, which is the number of times user i received rewards. The range of R_i is $1 \leq R_i \leq 3$, meaning that a user of high reliability counts as, at most, three users.

Figure 7 plots the changes in the number of validated triples of the cases with and without taking user reliability taken into consideration. In these simulation runs, user i's accuracy ($p_{\mathrm{acc}}(i)$) was set to a uniform random number between 0.6 and 1.0, and the user's confidence was set to a uniform random number between 0.6 and 1.0. The three strategies mentioned in Sec. 5.2.1 were used. As seen in this chart, the number of game sessions required to validate triples was decreased by taking the user's reliability into consideration. Although it is not certain if it introduces adverse effects such as erroneously validating triples, the number of games necessary to validate triples can be decreased by introducing the user's reliability.

5.2.3. *Contribution of users*

We also ran simulation experiments to investigate the relationship between the user's accuracy/confidence values and the user's contribution to the validation process. We set the user's accuracy and confidence to a random number with uniform distribution between 0.6 and 1.0. Figure 8 shows the result of one simulation run with a *random* strategy. In this simulation run, the Spearman's rank coefficient

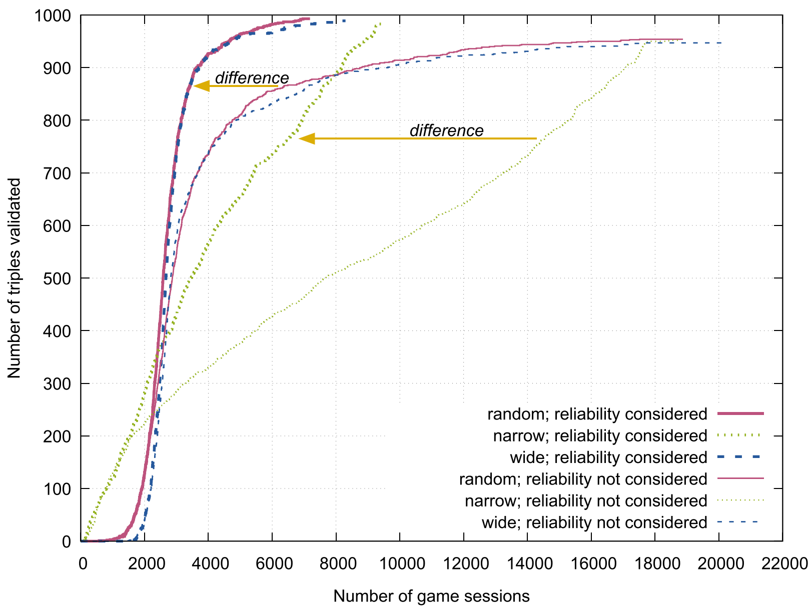


Fig. 7. Changes in the number of triples validated and the number of game sessions (with and without user reliability considered).

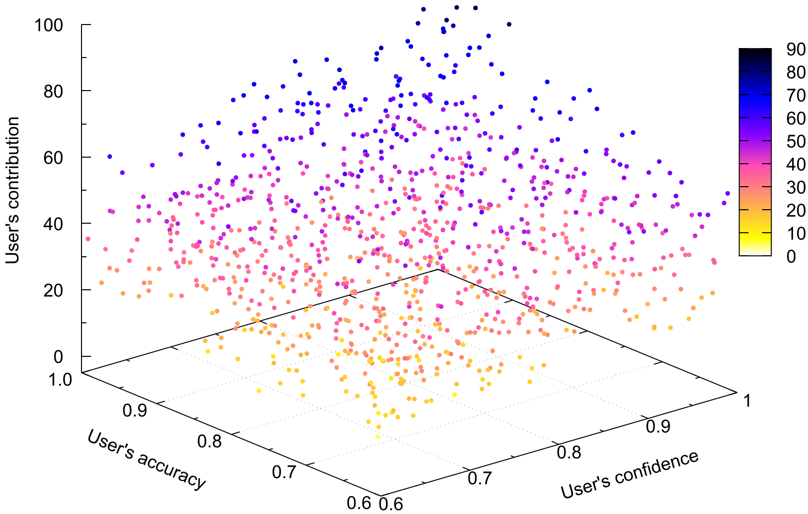


Fig. 8. Relationship between user's accuracy/confidence and user's contribution.

correlation between the user's accuracy and contribution was 0.392, and between the user's confidence and contribution it was 0.392. Figure 9 shows the relationship between the user's accuracy and the user's contribution, and Fig. 10 shows the relationship between the user's confidence and the user's contribution. As can be seen in these charts, generally the more accurate and confident a user is, the higher the user's contribution.

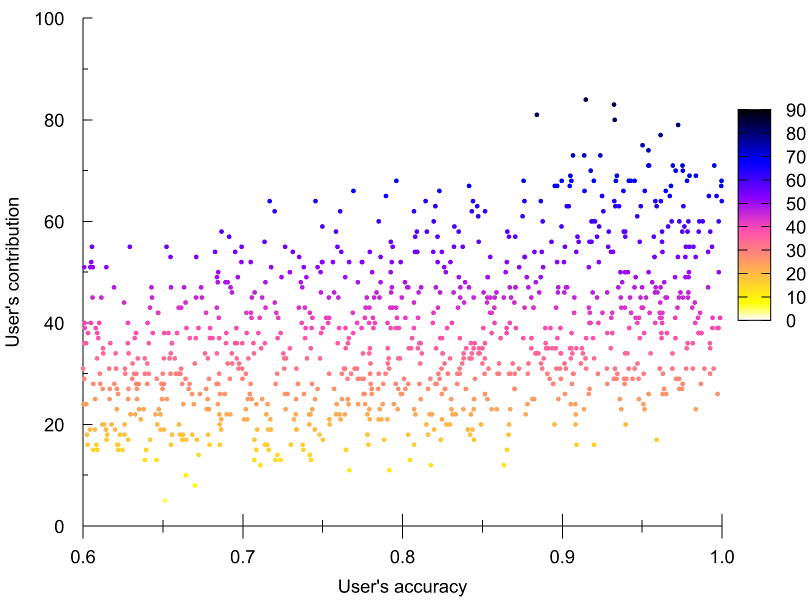


Fig. 9. Relationship between user's accuracy and user's contribution.

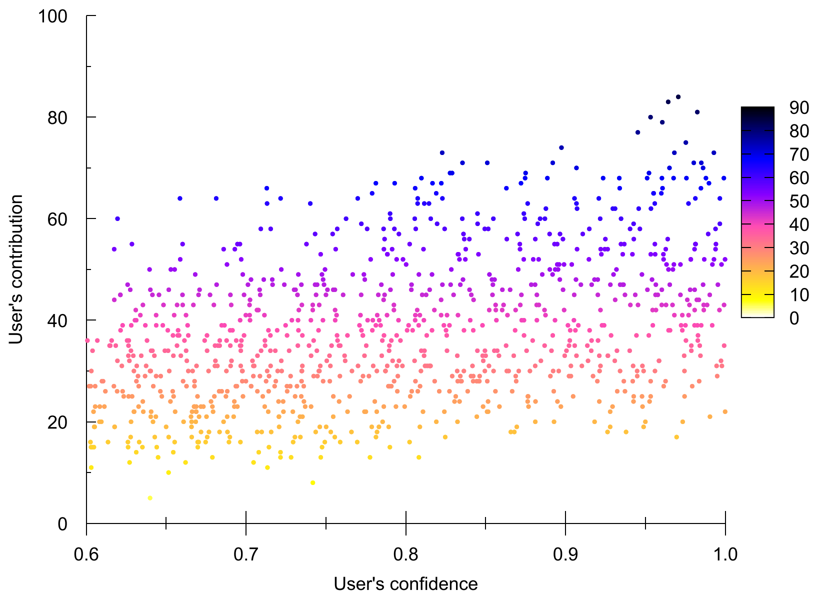


Fig. 10. Relationship between user's confidence and user's contribution.

Furthermore, we divided 1000 users into four groups with different accuracy and confidence values and ran a simulation to see what differences were introduced among the four groups in terms of the user's contribution. The user's accuracy and confidence values were set randomly according to the normal distribution with a mean of 0.9 (*high*) or 0.7 (*low*) and a standard deviation of 0.03. Four groups were composed based on the combination of median values of *high* and *low* in the user's accuracy and confidence values. Table 3 shows the average of three simulation runs of the maximum, median, and minimum of the user's contributions in each group. Additionally, Fig. 11 shows the distributions of the resulting user's contributions for each group for one simulation run. As these results indicate, a group with high

Table 3. Relationship between user's contribution and user's accuracy/confidence.

Group of users	Accuracy Confidence	*High* *High*	*High* *Low*	*Low* *High*	*Low* *Low*
User's contribution	Maximum	99.3	91	42.7	39.7
	Median	47.3	45	21	19
	Minimum	12.3	14	9	8.3
	Average	47.8	46.4	21.7	20.8

Notes: *High*: normal distribution with a mean of 0.9 and a standard deviation of 0.03.
Low: normal distribution with a mean of 0.7 and a standard deviation of 0.03.

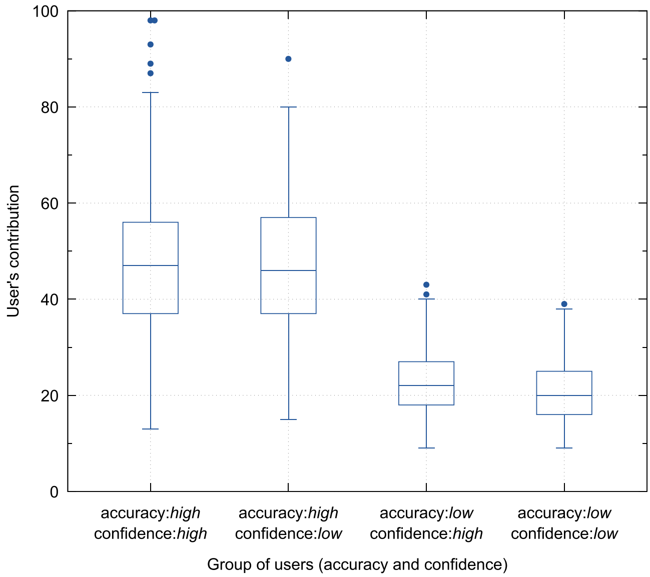


Fig. 11. Distribution of user's contribution of groups with different accuracy and confidence.

accuracy tends to make a higher contribution. The user's confidence also contributes to the user's contribution, but its effect is less than that of the user's accuracy.

6. Real-life Experiment

In addition to simulation experiments, we conducted a small experiment with human users using the prototype system implemented as a bot system (LINE bot) implemented on the LINE messaging service as described in Sec. 4.

6.1. *Consensus algorithm*

Naivechain, blockchain implementation used in the simulation experiments described above, does not implement a consensus algorithm such as Proof of Work (PoW) to validate blockchains in a distributed environment. From the viewpoint of examining the characteristics of the proposed knowledge validation process, it is not considered a major problem. However, in this real-life experiment, we added a simple PoW mechanism, as proposed in the bitcoin white paper.[25] In this PoW mechanism, a so-called miner generates a new block. A block generation task is controlled by a *difficulty* parameter. When a block is generated, a hash value is calculated. The value of *difficulty* specifies the condition that the generated hash value needs to be satisfied. In this experiment, we adopted a typical implementation where the value of *difficulty* specifies the number of leading zeros of a binary representation of the generated hash value. To obtain the hash value that satisfies the condition, an additional parameter (*nonce*) was included in the calculation of the hash value. *Nonce* was incremented by one until the desired hash value was obtained.

The value of *difficulty* parameter controls the mining rate, the speed of generating a block. In a public blockchain setting, *difficulty* is dynamically adjusted to control the mining rate. In this small experiment, we set the initial value of *difficulty* to 3 and adjusted it every 30 s by increasing or decreasing its value by 1 according to the rate of block generation so far.

Several other consensus algorithms have been proposed,[26] especially for a private blockchain; the PoW scheme may not be appropriate as it tends to incur considerable power consumption. The consensus algorithm needs to be selected according to the deployed environment.

6.2. *Method*

We created 40 triples to be validated in the experiment, part of which are shown in Table 4. Among them, 30 were *true* triples and 10 were *false* triples. These triples were chosen so that their truthfulness could be determined easily.

We recruited participants through social networking services, and 43 users joined the experiment. Most of them were university students who were active users of the LINE messaging service. We asked them to add the prototype chatbot system as a *friend* on the LINE messaging service. When the bot was added as a *friend*, the terms of the experiment were shown, which stated that a participant had the right to quit a game at any time without penalty and that no personal information would be stored in the log. A game was started only after the participant agreed with these terms.

The parameters used in the experiments were basically the same as those used in the simulation experiments described in Sec. 5. Namely, the threshold for judging a triple to be correct or not (S) was set to 10. The threshold ratio for judging the triple to be correct or not (α) was set to 0.8. The value of reward given was set to 100. We used the *narrow* strategy of selecting the next quiz because the number of triples validated can increase even when the number of games is low. We did not consider the user's reliability described in Sec. 5.2.2.

6.3. *Results and discussion*

When all the games ended by 43 participants, 33 triples were either validated or invalidated, amounting to 82.5% of all the triples prepared. Among 33 triples, 23 *true*

Table 4. Part of triples used in a real-life example experiment (translated from the Japanese originals).

Subject	*Predicate*	*Object*	Quiz sentence
Onion	Shape	Round	The shape of the onion is round.
Peach	Taste	Sweet	The taste of the peach is sweet.
Cabbage	Color	Black	The color of the cabbage is black.

triples were validated and 10 *false* triples were invalidated. All triples were correctly validated or invalidated; no *true* triples were judged *false* or vice versa.

We plotted the changes in the number of validated/invalidated triples according to the number of games sessions conducted as shown in Fig. 12. In addition to the total number of triples (in)validated, this figure also plots its breakdown of *true* and *false* triples. As shown in the figure, the overall tendency is the same as that of the simulation experiments.

As can seen in Fig. 12, after the certain number of game sessions the number of validated triples does not change because of the limited number of participants. The participants who joined the experiment later only answered the quizzes that were not judged to be correct or incorrect due to receiving few votes, and they received zero rewards. Note that if other participants joined later and the remaining triples were validated/invalided, they had a chance to receive rewards.

We examined the log of the 29 participants who received some rewards, excluding those who received no rewards as they joined the experiment later or answered DON'T KNOW to all quizzes. The average ratio of answering either YES or NO, that is, not answering DON'T KNOW, to the number of quizzes presented was 0.955, which corresponds to the *confidence* of a user model in the simulation experiments. The average ratio of correctly answering a question was 0.815, which corresponds to the *accuracy* of a user model. The Spearman's rank coefficient correlation between the participants' accuracy and rewards earned was 0.502. There was a moderate

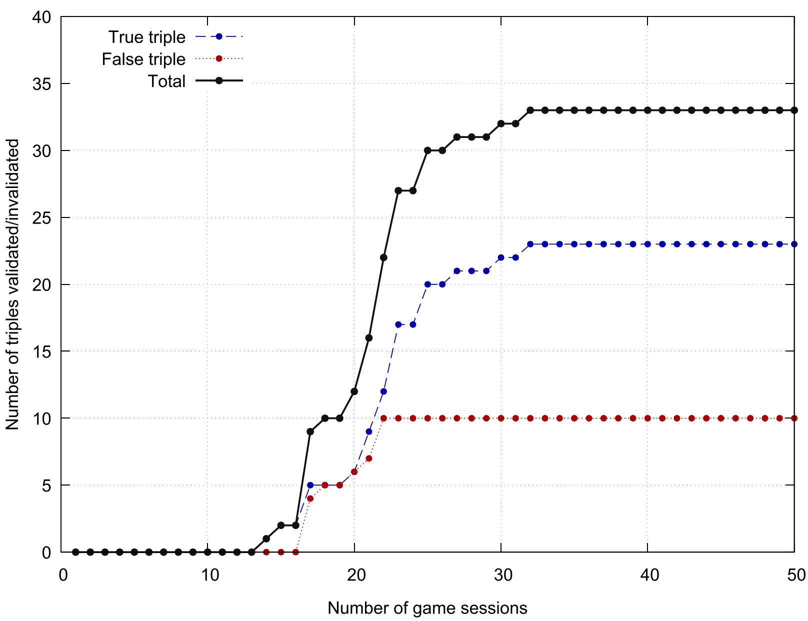


Fig. 12. Relationship between the number of game sessions and the number of triples validated/invalidated.

positive relationship between the participant's accuracy and rewards as in the simulation experiments. More than half of the participants had a *confidence* value of 1, and there was little variation in *confidence* values. For these reasons, we observed no particular correlation between the participants' confidence and rewards in this experiment. Overall, it can be said that triples can effectively be validated with human users using the proposed method.

7. Conclusion and Future Work

This paper described the use of gamified crowdsourcing for refining knowledge base. We adopted blockchain technology to ensure transparency of the user reward calculation. We also created a simple user model and conducted simulation experiments with many virtual users and three different methods for selecting the next task (quiz). These results indicate that the proposed approach can be used as a method for validating knowledge contents. In addition, the different strategies result in differences in the number of validated triples as the game sessions progress. We can select a suitable strategy for a given situation that is characterized by the number of triples to be validated and the number of users.

In this paper, we focused on the process of validating the knowledge contents already in the knowledge base. As there are many aspects in the knowledge base building, we plan to extend the proposed approach to the process of gathering knowledge contents by introducing other types of quizzes.

Furthermore, one of the benefits of blockchain technology is its distributed nature. Future work includes using it to distribute the knowledge refinement process over the network so that it can be easily scaled up.

Acknowledgment

This work was partially supported by JSPS KAKENHI Grant Number 18K11451.

References

1. H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, *Semant. Web* **8**(3) (2017) 489–508.
2. J. Howe, *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business* (Crown Business, 2009).
3. B. Morschheuser, J. Hamari, J. Koivisto and A. Maedche, Gamified crowdsourcing: Conceptualization, literature review, and future agenda, *Int. J. Hum.-Comput. Stud.* **106** (Suppl. C) (2017) 26–43.
4. L. von Ahn and L. Dabbish, Designing games with a purpose, *Commun. ACM* **51** (2008) 58–67.
5. G. Re Calegari, A. Fiano and I. Celino, A framework to build games with a purpose for linked data refinement, in *The Semantic Web — ISWC 2018*, eds. D. Vrandečić, K. Bontcheva, M. C. Suárez-Figueroa, V. Presutti, I. Celino, M. Sabou, L.-A. Kaffee and E. Simperl (Springer International Publishing, Cham, 2018), pp. 154–169.

6. L. von Ahn and L. Dabbish, Labeling images with a computer game, in *Proc. SIGCHI Conf. Human Factors in Computing Systems, CHI'04* (ACM, New York, NY, 2004), pp. 319–326.
7. A. M. Antonopoulos, *Mastering Bitcoin: Programming the Open Blockchain* (O'Reilly Media, 2017).
8. K. Kuwabara, T. Iwamae, Y. Wada, H.-H. Huang and K. Takenaka, Toward a conversation partner agent for people with aphasia: Assisting word retrieval, in *Intelligent Decision Technologies 2016: Proc. 8th KES Int. Conf. Intelligent Decision Technologies (KES-IDT 2016) — Part I*, eds. I. Czarnowski, M. A. Caballero, J. R. Howlett and C. L. Jain (Springer International Publishing, Cham, 2016), pp. 203–213.
9. D. Kurita, B. Roengsamut, K. Kuwabara and H.-H. Huang, Simulating gamified crowdsourcing of knowledge base refinement: Effects of game rule design, *J. Inf. Telecommun.* **2**(4) (2018) 374–391.
10. H. Bu and K. Kuwabara, Toward blockchain-assisted gamified crowdsourcing for knowledge refinement, in *Intelligent Information and Database Systems*, eds. N. T. Nguyen, K. Jearanaitanakij, A. Selamat, B. Trawiński and S. Chittayasothorn (Springer International Publishing, Cham, 2020), pp. 3–14.
11. Y. Lu, The blockchain: State-of-the-art and research challenges, *J. Ind. Inf. Integr.* **15** (2019) 80–90.
12. M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xiang and R. H. Deng, CrowdBC: A blockchain-based decentralized framework for crowdsourcing, *IEEE Trans. Parallel Distrib. Syst.* **30**(6) (2019) 1251–1266.
13. S. Zhu, Z. Cai, H. Hu, Y. Li and W. Li, zkCrowd: A hybrid blockchain-based crowdsourcing platform, *IEEE Trans. Ind. Inf.* **16**(6) (2020) 4196–4205.
14. T. Min, H. Wang, Y. Guo and W. Cai, Blockchain games: A survey, in *2019 IEEE Conf. Games (CoG)* (IEEE, 2019), pp. 1–8.
15. J. Hu, K. Yang, K. Wang and K. Zhang, A blockchain-based reward mechanism for mobile crowdsensing, *IEEE Trans. Comput. Soc. Syst.* **7**(1) (2020) 178–191.
16. X. Lin, J. Li, J. Wu, H. Liang and W. Yang, Making knowledge tradable in edge-AI enabled IoT: A consortium blockchain-based efficient and incentive approach, *IEEE Trans. Ind. Inf.* **15** (2019) 6367–6378.
17. J. D. Harris and B. Waggoner, Decentralized and collaborative AI on blockchain, in *2019 IEEE Int. Conf. Blockchain* (IEEE, 2019), pp. 368–375.
18. H.-G. Fill and F. Häerer, Knowledge blockchains: Applying blockchain technologies to enterprise modeling, in *Proc. 51st Hawaii Int. Conf. System Sciences* (2018), pp. 4045–4054. http://hdl.handle.net/10125/50398.
19. G. Nyame, Z. Qin, K. O.-B. Obour Agyekum and E. B. Sifah, An ECDSA approach to access control in knowledge management systems using blockchain, *Information* **11**(2) (2020), https://doi.org/10.3390/info11020111.
20. O. Scekic, S. Nastic and S. Dustdar, Blockchain-supported smart city platform for social value co-creation and exchange, *IEEE Internet Comput.* **23** (2019) 19–28.
21. M. Sopek, P. Gradzki, W. Kosowski, D. Kuziski, R. Trójczak and R. Trypuz, GraphChain: A distributed database with explicit semantics and chained RDF graphs, in *Companion Proc. Web Conf. 2018, WWW'18* (International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2018), pp. 1171–1178.
22. S. Wang, C. Huang, J. Li, Y. Yuan and F. Wang, Decentralized construction of knowledge graphs for deep recommender systems based on blockchain-powered smart contracts, *IEEE Access* **7** (2019) 136951–136961.

23. M. Emaldi, K. Zabaleta and D. López-de Ipiña, AUDABLOK: Engaging citizens in open data refinement through blockchain, *Proceedings* **31**(1) (2019), https://doi.org/10.3390/proceedings2019031052.
24. T. Iwamae, K. Kuwabara and H.-H. Huang, Toward gamified knowledge contents refinement — case study of a conversation partner agent, in *Proc. 9th Int. Conf. Agents and Artificial Intelligence (ICAART 2017)*, Vol. 1 (SCITEPRESS (Science and Technology Publications, Lda), 2017), pp. 302–307.
25. S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System* (2008), https://nakamotoinstitute.org/literature/bitcoin/.
26. S. M. H. Bamakan, A. Motavali and A. Babaei Bondarti, A survey of blockchain consensus algorithms performance evaluation criteria, *Expert Syst. Appl.* **154** (2020) 113385.