Simulating gamified crowdsourcing of knowledge base refinement: effects of game rule design*

Daiki Kurita\textsuperscript{a}, Boonsita Roengsamut\textsuperscript{a}, Kazuhiro Kuwabara\textsuperscript{b} and Hung-Hsuan Huang\textsuperscript{b}

\textsuperscript{a}Graduate School of Information Science and Engineering, Ritsumeikan University, Kusatsu Shiga, Japan; \textsuperscript{b}College of Information Science and Engineering, Ritsumeikan University, Kusatsu Shiga, Japan

\textbf{ABSTRACT}

This paper discusses a gamification design for knowledge base refinement. We present a simulation model for game playing and examine the possibility of estimating the effects of game rule design. The maintenance of a knowledge base involves human intervention and is one of the application domains of human computation. Using the concept of crowdsourcing, refinement tasks can be delegated to many casual users over a network. In addition, gamification, such as games with a purpose (GWAP), is a useful idea to motivate workers in crowdsourcing by making tasks enjoyable. For effective gamification, designing the game rules is critical. In this paper, we present a model for simulating the gamified knowledge base refinement process to estimate the effects of different game rule designs beforehand. We use a rental apartment FAQ system as an example knowledge base and show that by invoking game playing when a problem is found in a knowledge base, it can be properly updated. In addition, we discuss the effectiveness of using a calibration quiz to estimate the reliability of users.

\textbf{ARTICLE HISTORY}

Received 5 May 2017
Accepted 30 October 2017

\textbf{KEYWORDS}

Crowdsourcing; gamification; knowledge base refinement; linked data

\textbf{1. Introduction}

The web has evolved into a platform on which knowledge content is created and shared. Such technologies as data mining or machine learning have made significant advancements for extracting meaningful data from large data. However, constructing a high-quality knowledge base still requires human participation, especially from domain experts. Hiring domain experts, however, can be expensive. Thus, how to utilize the power of many casual users is crucial. Crowdsourcing is one notable framework to tackle this problem (Doan, Ramakrishnan, & Halevy 2011; Howe 2009).

In crowdsourcing, maintaining the motivation of many users plays a crucial role, and the importance of intrinsic motivation has been identified (Ryan & Deci 2000). The concept of gamification, which is a popular method for providing intrinsic motivation and engaging users, converts a non-gaming context into a playful game so that users are more actively
engaged in a given task (Deterding, Dixon, Khaled, & Nacke 2011). It is also utilized in crowdsourcing (Morschheuser, Hamari, & Koivisto 2016).

A crowdsourcing task is often divided into microtasks called human intelligence tasks (HITs), each of which is assigned to a worker. In a typical crowdsourcing framework, such as Amazon Mechanical Turk (MTurk), an external reward (often monetary) is given to workers. If we turn a microtask into a playful game, an intrinsic motivator to enjoy it might be used to reward workers instead of an external reward. One such an example is games with a purpose (GWAP) (von Ahn & Dabbish 2008), where the original task is indirectly executed as the side benefits of playing the game.

GWAP is also used in semantic knowledge acquisition (Šimko & Bieliková 2014; Siorpaes & Hepp 2008). For example, SpotTheLink, which is targeted for ontology alignment (Thaler, Simperl, & Siorpaes 2011), was developed as part of the OntoGame series. Gamification is also applied in the maintenance of knowledge bases (Jovanovic 2015). In addition, a web-based game collected commonsense knowledge for understanding goals in everyday life (Lieberman, Smith, & Teeters 2007). Factual knowledge in rule form was also collected through a game (Rodosthenous & Michael 2016).

When the gamification approach is applied to knowledge base refinement, designing a game is an important issue for obtaining high-quality results, especially since it is often difficult to assess the effects of particular game rules before the game is actually played with human users. To tackle this problem, we made a simulation model of the game execution. By simulating the game execution, we expect to determine the effectiveness of the game rule design beforehand and adjust the rules before letting human users play it.

As the first step towards this goal, we consider a model that executes a game for refining a knowledge base, which was constructed as linked data (Bizer, Heath, & Berners-Lee 2009). Information necessary to update the linked data are collected through game execution. Using this model, we estimate and compare the effects of different game rule designs on the performance of knowledge refinement (Kurita, Roengsamut, Kuwabara, & Huang 2016).

Kurita et al. (2016) considered a simple model where the weights on the links were directly updated. In this paper, we extend this model to handle a more generalized situation. For example, in the previous model, a quiz only handled one link. In this paper, a quiz is extended to simultaneously handle multiple links. In addition, in the original paper, we considered a case where just one problem was found and fixed. In this paper, we examine a case where multiple problems are simultaneously fixed.

The rest of this paper is organized as follows. The next section describes related works. Section 3 presents the knowledge base we used, and Section 4 describes the game designed for this paper. Section 5 presents a model to simulate the knowledge base refinement process and describes our experimental results. The final section concludes this paper and discusses future work.

2. Related work

Engaging users (workers) in the task of refining a knowledge base is important to guarantee the quality of the results. Gamification is an effective way to enhance user incentives. A model for appropriate incentive design was proposed (Feyisetan, Simperl, Van Kleek, &
Shadbolt (2015) where a combination of gamification and paid microtasks was investigated, and a predictive model was created for estimating a proper set of worker incentives.

Although gamification enhances the motivation of users, it does not necessarily lead to high-quality output (Juźwin et al. 2014). Games with a purpose (GWAP) rely on a clever game rule design to obtain correct data (von Ahn & Dabbish 2008). In the ESP game, which is one typical example of GWAP, the target task is to put appropriate labels on an image (von Ahn & Dabbish 2004), and two independent players separately label a given image and receive points when their labels match. Since no communication is assumed between the two players, the best strategy for a player is to produce correct labels as much as possible. In this way, more appropriate results will probably be obtained. However, designing such game rules is not necessarily easy for various domains.

To obtain high-quality results, qualified workers are preferable and engaging them in the task is also important. Quizz is a gamified crowdsourcing platform (Ipeirotis & Gabriilovich 2014), where the competence of workers is estimated using a calibration quiz, a special type of quiz whose answers are known beforehand. The system successfully collects and curates correct knowledge. Using this platform, dynamic task allocation was examined by predicting the survival probability of workers to indicate the probability that they will proceed to the next task offered by the system (Kobren, Tan, Ipeirotis, & Gabriilovich 2015).

These works focus more on the detailed analysis of the behaviour of crowdsourcing systems. In contrast, in this paper we simulate game execution to estimate the effects of a particular game rule design beforehand.

3. Rental apartment FAQ

3.1. Overview

We use the knowledge base of a rental apartment’s frequently asked question (FAQ) system (Saito, Roengsamut, & Kuwabara 2014) as an example knowledge base. This application, which was originally developed to support international students living in Japanese rental apartments, contains some troubleshooting knowledge about various problems that might occur, such as faulty air conditioners. Its knowledge base is represented as linked data (Bizer et al. 2009), more specifically, resource description framework (RDF), developed in the Semantic Web to facilitate sharing knowledge on the web.

The knowledge base contains a simple domain ontology of the apartment, including a typical floor plan and common fixtures or equipment in it. Using the domain ontology, a link is derived between an FAQ entry and its most closely related section of a floor plan. We extended the original FAQ knowledge base to include the link’s weight to represent the strength of relationships. Figure 1 shows that a question statement of an FAQ entry, There’s no hot water in my shower, is linked to different floor plan sections (kitchen and bathroom) with different weights. The values of the weights are calculated from the domain ontology, as explained in Section 3.3.

There are different ways to put a weight to a link in the RDF model, for example, extending a triple-based RDF model to a quadruplet to include the weight of each link (Cedeño & Candan 2011). In this extension, SPARQL, which is query language for RDF, is also extended so that queries can be made for links with weights. In this paper, however, we did not
change the underlying data model to use a widely available RDF database software. Instead, we used the classical reification technique to store the link’s weight (Figure 1).

3.2. Domain ontology

The domain ontology mainly contains words corresponding to various equipment or fixtures often found in an apartment, such as an air conditioner or a shower. We treat such terms as *keywords*. The domain ontology also contains words that describe a *floor plan section*, such as *living room* or *kitchen*. The relationships between keywords that represent various equipment or fixtures and floor plan sections are also described in the domain ontology. This knowledge derives a link between a question statement and a relevant floor plan section.

3.3. Derivation of a link

When a link is derived from a question statement to the floor plan section, its weight is calculated and put on the link. A question statement, or an FAQ entry, may be associated with multiple places (floor plan sections). In such a case, it is preferable to represent the strength of each association. Since the link between a question statement and a floor plan section is derived from the relationship between things like fixtures or equipment represented by a keyword and a floor plan section, their relationship is also extended to have a particular weight.

Let $W_{q,s}$ denote the weight of the relationship between question statement $q$ and floor plan section $s$. $W_{q,s}$ is calculated as follows:

$$W_{q,s} = \sum_{k \in K(q)} e_{q,k} \cdot w_{k,s},$$

where $K(q)$ denotes a set of keywords representing equipment or fixtures associated with question statement $q$, $e_{q,k}$ represents the importance of keyword $k$ in question statement $q$, and $w_{k,s}$ represents the weight of the relationship between keyword $k$ and floor plan section $s$. 

![Figure 1. Example link between question statement and floor plan sections.](image)
The most relevant floor plan section of question statement $q$, $s^*$, can be obtained:

$$s^* = \arg \max_s W_{q,s}.$$  \hspace{1cm} (2)

An example domain ontology with a link weight is depicted in Figure 2. Here, shower is connected to kitchen and bathroom with link weights of 10 and 60, where shower and hot_water are treated as keywords, and kitchen and bathroom are treated as floor plan sections: $w_{\text{shower,kitchen}} = 10$ and $w_{\text{shower,bathroom}} = 60$. hot_water is also connected to kitchen and bathroom but with different weights: $w_{\text{hot_water,kitchen}} = 40$, $w_{\text{hot_water,bathroom}} = 40$. The link’s weight represents the closeness of the relationships.

Next consider question statement $q_1$ (There’s no hot water in my shower), which contains shower and hot_water keywords. Based on the weight of the links between these keywords and the floor plan sections, the weights of the links between question statement $q$ and floor plan section $s$ (bathroom or kitchen) are calculated:

$$W_{q_1,\text{bathroom}} = e_{q_1,\text{hot_water}} \cdot W_{\text{hot_water,bathroom}} + e_{q_1,\text{shower}} \cdot W_{\text{shower,bathroom}},$$  \hspace{1cm} (3)

$$W_{q_1,\text{kitchen}} = e_{q_1,\text{hot_water}} \cdot W_{\text{hot_water,kitchen}} + e_{q_1,\text{shower}} \cdot W_{\text{shower,kitchen}}.$$  \hspace{1cm} (4)

Basically, $e_{q,k}$ represents the effect of keyword $k$ in question statement $s$. Intuitively, if keyword $k$ has a relationship with only one floor plan section, its effect is large. Conversely, if keyword $k$ has relationships with many floor plan sections, its effect is small. $h(k)$ denotes the degree of such effects. $h(k)$ is generally set to 1, but we set it to 2 for keywords that correspond to fixture or equipment because such keywords are more likely concerned with a particular place in an apartment. Whether the keyword has such a property is determined based on the domain ontology. For example, Figure 2 shows that shower (depicted as word:shower) corresponds to fixture (depicted as c:fixture). Continuing the example above, based on this heuristics, we set the values of $h(k)$ as follows: $h(\text{shower}) = 2$ and $h(\text{hot_water}) = 1$.

![Figure 2. Domain ontology with a link weight.](image-url)
Using \( h(k) \), \( e_{q,k} \) can be calculated:

\[
e_{q,k} = \frac{h(k)}{\sum_{k' \in K(q)} h(k')}. \tag{5}
\]

Thus,

\[
e_{q_1,\text{hot water}} = 1/(1 + 2) = 0.33, \tag{6}
\]

\[
e_{q_1,\text{shower}} = 2/(1 + 2) = 0.67. \tag{7}
\]

Using these values, the weight of the links between question statement \( q \) and a floor plan section can be calculated:

\[
W_{q_1,\text{bathroom}} = e_{q_1,\text{hot water}} \cdot w_{\text{hot water,bathroom}} + e_{q_1,\text{shower}} \cdot w_{\text{shower,bathroom}} \tag{8}
\]

\[
= 0.33 \cdot 40 + 0.67 \cdot 60 = 53.33, \tag{9}
\]

\[
W_{q_1,\text{kitchen}} = e_{q_1,\text{hot water}} \cdot w_{\text{hot water,kitchen}} + e_{q_1,\text{shower}} \cdot w_{\text{shower,kitchen}} \tag{10}
\]

\[
= 0.33 \cdot 40 + 0.67 \cdot 10 = 20. \tag{11}
\]

Since \( W_{q_1,\text{bathroom}} > W_{q_1,\text{shower}} \), the most relevant floor plan section of question statement \( q_1 \) is bathroom.

### 3.4. Adding a new question

The rental apartment FAQ system has a function to add a new question with a photo that deepens the question’s description. A question statement can also include hashtags. Since hashtagged words can be treated as keywords, we can eliminate the keyword extraction process.

For example, a question statement, **There’s no hot water in my shower**, can be input as the following text: **There’s no #hot_water in my #shower**. From this text, **hot_water** and **shower** can be extracted as keywords. When a question is presented to other users, the hashtag itself is omitted and the question is displayed as: **There’s no hot water in my shower**.

If a photo is uploaded with a floor plan section, a correct link of the accompanying question statement is given. In this case, only from the question statement text, the most related floor plan section is calculated using the domain ontology. If the result is identical as the place where the photo was taken, the domain ontology is acceptable. If not, an error correction procedure will be invoked. Since we manage the link with weights, the threshold of the weight with which an error is judged needs to be carefully set.

### 4. Game design

Here we consider a game through which to refine the contents of a knowledge base. Game playing is invoked when a problem is found in the knowledge base. For example, if the most relevant floor plan section calculated from the link weights of the domain ontology does not match the one that was manually input, we assume that the data in
the domain ontology may not be correct. In such a case, game playing is invoked to gather the information necessary to fix the problem (Figure 3). The game is extended from the Aparto Game developed for gamifying knowledge base construction of a rental apartment FAQ system (Roengsamut, Kuwabara, & Huang 2015).

4.1. Game for knowledge base correction

When an error is found in the knowledge base, we collect the data required to fix it through game playing. For example, suppose that the most related floor plan section of the question statement There’s no hot water in my shower is calculated as kitchen using the data in the knowledge base. This is not intuitively correct. Since the weight of a link between a question statement and the floor plan section is calculated from the weights of a link between a keyword and a floor plan section, these weights might not be appropriate.

Kurita et al. (2016) generated a new game that presented a quiz to ask users directly about the relationship between a keyword and a floor plan section. For example, a user (player) is asked which is the most relevant floor plan section of shower and kitchen, bathroom, living_room, and toilet are presented as possible answer choices. A user selects the answer from these choices.

Instead, in this paper, we consider a more generalized case, where a user is asked about the relationship between a floor plan section and a question statement that might contain multiple keywords so that a greater variety of games is possible. In addition, since a new question statement does not need to be generated specifically for a game, the game-playing process can more easily be blended into the main use of the system.

To gather the information to update the value of the link’s weight, we created the following quiz as a game. The quiz shows a question statement and asks users to select the most relevant floor plan section (Figure 4).

After the user input is obtained, the weights of the links are updated:

$$\Delta W_{q,s} = \begin{cases} \alpha & (s = a), \\ -\frac{\alpha}{N_q - 1} & (s \neq a), \end{cases}$$

(12)

where \(a\) represents the user’s answer (the selected floor plan section) of the quiz, and \(\alpha\) is a coefficient of the weight updates. Here, \(N_q\), which represents the number of possible

\[\text{Figure 3. Knowledge base refinement overall flow.}\]
choices of the quiz, is determined as the number of floor plan sections whose \( W_{q,s} \) value is positive:

\[
N_q = |s| W_{q,s} > 0 |.
\] (13)

Using \( \Delta W_{q,s} \), the weight of a link between a keyword and a floor plan section, \( w_{k,s} \), is updated by distributing \( \Delta W_{q,s} \) over \( w_{k,s} \):

\[
w_{k,s} \leftarrow \begin{cases} 
  w_{k,s} + \Delta W_{q,s} \cdot \frac{e_{q,k} \cdot w_{k,s}}{\sum_{k' \in K(q)} e_{q,k'} \cdot w_{k',s}} & (s = a), \\
  w_{k,s} - \Delta W_{q,s} \cdot \frac{e_{q,k} \cdot w_{k,s}}{\sum_{k' \in K(q)} e_{q,k'} \cdot w_{k',s}} \cdot \frac{1}{N_q - 1} & (s \neq a).
\end{cases}
\] (14)

Note that when \( N_q = 1 \), no update is performed, since the most relevant floor plan section is narrowed down to one section. When \( \sum_{k' \in K(q)} e_{q,k'} \cdot w_{k',s} \) becomes zero, the update will not be performed either, since it is impossible for question statement \( q \) to be related to floor plan section \( s \). After updating \( w_{q,s} \), the weight of the link between question statement \( q \) and floor plan section \( s \), \( W_{q,s} \), is recalculated.

### 4.2. Assessing a user’s reliability

When a knowledge base is being corrected, the effects of unreliable data from users must be removed. We assess user reliability by using a calibration quiz whose correct answers are known beforehand. This calibration quiz is mixed into a series of main quizzes. Based on its answers, user reliability \( R \) is estimated. With \( R \), Formula (12) is rewritten:

\[
\Delta W_{q,s} = \begin{cases} 
  c(R) \cdot \alpha & (s = a), \\
  -c(R) \cdot \frac{\alpha}{N_q - 1} & (s \neq a),
\end{cases}
\] (15)

where \( c(R) \) is a function that returns the value of the coefficient of updating a link by a user whose reliability is given as \( R \).

### 5. Experiment and evaluation

We conducted simulation experiments to confirm that the knowledge base can be refined by the proposed game. The simulation experiments focused on the changes in the weight of a link between a question statement and its most relevant floor plan section after the game presented in Section 4.1 is executed many times. In addition, we confirmed that the changes in the game rule design can be examined.
The simulation experiments include two cases. One is targeted for a quiz with multiple keywords (Section 5.1), and the other is targeted for multiple quizzes executed together (Section 5.2).

5.1. Experiments with multiple keywords

5.1.1. Purpose

Kurita et al. (2016) assumed that a quiz is only concerned with one keyword. In the first case of simulation experiments, we examined whether knowledge contents can be refined even when a quiz contains multiple keywords. We also examined the effect of a calibration quiz.

5.1.2. Method

In this part of our simulation experiments, we addressed a situation where the initial values of the link weights are assumed to be erroneously set (Table 1) and examined whether these link weights can be corrected through game playing. Table 1 also includes the weights of question statement $q_1$ (There’s no hot water in my shower) for each floor plan section calculated from the link weights, assuming that $e_{q_1,\text{hot-water}} = 0.33$ and $e_{q_1,\text{shower}} = 0.67$. The most relevant floor plan section is kitchen based on the given link weights. This is not intuitively correct; it should be bathroom. The simulation experiment is intended to determine whether the link weights can be updated so that the most relevant floor plan section becomes bathroom.

In this simulation experiment, we considered the four floor plan sections shown in Table 1. Since the values of $W_{q_1,s}$ are all positive for these floor plan sections, $N_{q_1}$ is set to 4. The main quiz asks a user to select the most related floor plan section of the question statement from the following list: kitchen, bathroom, living_room, and toilet.

The calibration quiz, which estimates user reliability, has a list of four choices, one of which is the correct answer. We assume two kinds of users: reliable user $U_R$ and non-reliable user $U_N$. Reliable user’s reliability $R$ is set to 1, and that of the non-reliable user is set to 0.

We assume that a reliable user will answer the calibration quiz correctly with a 0.92 probability and select an answer for the main quiz from the choice list of kitchen, bathroom, living_room, and toilet with probabilities of 0.02, 0.92, 0.04, and 0.02, respectively. A non-reliable user is assumed to answer the calibration quiz randomly, meaning that the probability of selecting a correct answer is 0.25, since there are four answer choices. We also assume that non-reliable users answer the main quiz by selecting from the choices with equal probability.

We assume a large number of users. Each game execution consists of randomly selecting one user (from a large body of users) who plays the game. This game round is repeated...

| Table 1. Initial link weights between keywords and floor plan sections used in simulations and initially calculated weights between a question statement and floor plan sections. |
|---|---|---|
| **Floor plan section** | **Keywords** | **Question statement** |
|  | hot_water | shower | $q_1$ (There’s no hot water in my shower) |
| kitchen | 40 | 80 | 66.67 |
| bathroom | 40 | 30 | 33.33 |
| living_room | 20 | 20 | 20.00 |
| toilet | 40 | 20 | 26.67 |
500 times. As the game progresses, the link weights between the keywords and floor plan sections are updated, and \( W_{q_1,s} \) may become zero for some floor plan sections, which indicates that the question statement has almost no relationship with that floor plan section. In such a case, that floor plan section is removed from the answer choices. When there is only one floor plan section whose weight exceeds zero (that is, \( N_{q_1} \) becomes 1), the link weights are no longer updated since \( N_{q_1} = 1 \) means that the most relevant floor plan section has been determined.

### 5.1.3. Results and discussion

We set the value of coefficient \( \alpha \) in Formula (12) to 0.6 based on the results of a preliminary experiment. In addition, \( c(R) \) in Formula (15) was set to 2.0 for \( R=1 \) (reliable user), and for \( R=0 \), \( c(0) = 1/c(1) = 0.5 \) (non-reliable user) so that the reliable user’s answer has a greater effect on updating link weights. Three different user sets with different reliable user percentages were considered; the probabilities of reliable users \( P(U_R) \) were set to 0.9, 0.6, and 0.3. The changes in the weight of a link from question statement \( q_1 \) (There’s no hot water in my shower) to floor plan sections kitchen and bathroom in one trial run are plotted in Figure 5, where the x-axis represents the number of game rounds that were played. The plots also include both cases with/without the calibration quiz.

As shown in the graphs in Figure 5, the weight of the question statement to kitchen decreases, but the weight of the question statement to bathroom increases for all three cases of different probabilities of a reliable user. At some point, the weight for bathroom exceeds that of kitchen. This means that the most related floor plan section of the question statement becomes bathroom after enough game rounds are executed.

These simulation results indicate that an error found in the knowledge base can be fixed by game playing. In addition, when the calibration quiz is utilized, the intersection of the curves of the weights of two links comes earlier in the game rounds than without the calibration quiz, suggesting its effectiveness.

To examine the effects of the value \( c(1) \), we run additional simulations under identical conditions except that the value of \( c(1) \) was doubled (\( c(1) = 4.0 \)), and the value of \( c(0) \) accordingly became (\( c(0) = 1/c(1) = 0.25 \)). The results are shown in Figure 6. With a larger \( c(1) \), the most relevant floor plan section was changed more quickly with a calibration quiz. This indicates that we can control the effects of a calibration quiz by properly setting the value of \( c(1) \).

In addition, we quantitatively examined the effects of the calibration quiz by examining the number of games when two weight curves intersect (intersection point) and defined the improvement rate of the calibration quiz \( GR \) as the ratio of the number of games at the intersection point with a calibration quiz \( (G_w) \) to the number of games at the intersection point without it \( (G_{wo}) \). That is, \( GR = G_w / G_{wo} \). We varied the probabilities of reliable users from 0.3 to 1.0 as well as the effects of reliable users by setting \( c(1) = 2 \) and \( c(1) = 4 \). For each case, we repeated each simulation 1,000 times and calculated the average improvement rate. The results are plotted in Figure 7, which also includes the average number of games at the intersection point. The improvement rate does not vary even if the probability of the reliable users varies, although the number of games required to change the most relevant floor plan section is larger with smaller probability of reliable users.
Figure 5. Changes in link weights with/without calibration quiz in one trial run ($c(1) = 2.0$). (a) Probability of reliable users: $P(U_R) = 0.3$, (b) probability of reliable users: $P(U_R) = 0.6$ and (c) probability of reliable users: $P(U_R) = 0.9$. 
Figure 6. Changes in link weights with/without calibration quiz in one trial run \( c(1) = 4.0 \). (a) Probability of reliable users: \( P(U_R) = 0.3 \), (b) Probability of reliable users: \( P(U_R) = 0.6 \) and (c) Probability of reliable users: \( P(U_R) = 0.9 \).
5.2. Experiments with multiple quizzes

5.2.1. Purpose
In this part of the simulation experiments, we examine a case with multiple question statements that are being refined at the same time. More specifically, we create a quiz from each question statement and run simulation experiments to see if multiple question statements can be refined in parallel with the created multiple quizzes. In addition, we examine how to select a quiz from a group of candidates.

5.2.2. Method
To examine a case where multiple question statements exist, we added another question statement (q2): *There’s no hot water in the dishwasher*. This question statement has two keywords: *hot_water* and *dishwasher*. We assume that the domain ontology defined *dishwasher* as *equipment*, meaning that the weight coefficient of *dishwasher*, $h($dishwasher$)$, is set to 2. The initial weights in the simulation experiments are shown in Table 2. They are identical with the preceding experiments with multiple keywords in Section 5.1 except that values for *dishwasher* are added. In this table, keyword *dishwasher* erroneously has

![Figure 7](image-url): Improvement rate and intersection points vs. probability of reliable users.

| Floor plan section | Keywords | Question statement |
|--------------------|----------|--------------------|
| shower | 66.67 | 33.33 |
| toilet | 20.00 | 20.00 |

Table 2. Initial link weights between keywords and floor plan sections used in simulation with multiple quizzes.
a bigger weight with bathroom, but it should have a bigger weight with kitchen since a dishwasher is predominantly placed in the kitchen.

Using these initial weights, the most relevant floor plan section of question statement $q_2$ is calculated as bathroom, which is not correct. Thus, in this example, there are two question statements whose most relevant floor plan section is wrong.

\textbf{Figure 8.} Changes in link weights when two question statements simultaneously updated in one trial run. (a) Probability of reliable users: $P(U_R) = 0.6$ and (b) Probability of reliable users: $P(U_R) = 0.9$. 
In this experiment, we used a calibration quiz since we confirmed their effect in the preceding experiments (Section 5.1). We also assume many users, and in each game round, we select a user from them and present a quiz randomly selected from two quizzes with certain probability. Each of two quizzes ($Q_1$ and $Q_2$) corresponds to a particular question statement: $q_1$ or $q_2$. Since there are two quizzes in this experiment, we repeated 1000 game rounds, twice that of the preceding experiments. We set the

![Figure 9](image_url)

**Figure 9.** Intersection points vs. mixing rate of quizzes. (a) Probability of reliable users: $P(U_R) = 0.6$ and (b) Probability of reliable users: $P(U_R) = 0.9$. 
value of $\alpha$ to 0.6 and $c(1)$ to 4.0. We also determined how reliable and unreliable users answered quizzes (including a calibration quiz) in the same manner as in the preceding experiments.

5.2.3. Results and discussion

Figure 8 shows changes of the weights of the link of two target question statement $q_1$ and $q_2$ in one trial run with the probability of reliable users $P(U_R)$ set to 0.6 (Figure 8(a)) and 0.9 (Figure 8(b)). Here, the probability of quiz $Q_1$ selected is set to 0.6. In these graphs, intersection points are marked with a circle. Even when two question statements were updated simultaneously, the link weights were correctly updated. The higher probability of reliable users led to the faster refinement. In addition, question statement $q_1$ was updated faster since the corresponding quiz $Q_1$ was asked more times than quiz $Q_2$, which corresponds to question statement $q_2$.

To examine the effects of different probabilities of quizzes, we varied probabilities of quizzes (mixing rate of quizzes). For each case, we repeated the simulation 1000 times and calculated the average number of games at intersection points (Figure 9).

Figure 9(a) shows the results with the probability of reliable users set to 0.6 ($U(R) = 0.6$). When the percentage of quiz $Q_1$ for question statement $q_1$ is big, the number of games at intersection points for $q_1$ is small, meaning that the knowledge content refinement can be performed more quickly. To examine the overall performance, we also plotted the sum of average number of games at intersection points of two question statements $q_1$ and $q_2$. The sum is the lowest when two quizzes are mixed with equal probabilities. This result suggests that when multiple quizzes are involved, it is better to mix them evenly.

Figure 9(b) shows the results with $U(R) = 0.9$. The number of games at intersection points were generally smaller than the case where $U(R) = 0.6$, which means that the greater the probability of reliable users, the better the results are. In addition, the probability of quiz $Q_1$ of 0.5 also produced the best results for the $U(R) = 0.9$ case. This result indicates that the percentage of reliable users will not affect the best mixing rate of the quizzes.

6. Conclusion and future work

This paper presented a gamified crowdsourcing approach towards knowledge base refinement and a simulation experiment to examine the effects of game rule design. Using a rental apartment FAQ system as an example knowledge base, we created an execution model of the game for its knowledge base refinement focusing on the effects of a calibration quiz intended for estimating user reliability.

Compared with our previous work (Kurita et al. 2016), we conducted simulations under more generalized conditions. That is, a quiz can have not only one but multiple keywords. This made it possible to use a question statement as a quiz in the target application. In addition, we simulated a case where multiple issues in the knowledge contents are simultaneously refined. The simulation results indicate that through game playing, the weight of the links in knowledge bases can be updated and that estimating user reliability accelerates knowledge base refinement. Although we ran simulations with limited examples, this simulation approach has potential as a tool to investigate the outcome of gamified crowdsourcing. Further analysis is necessary to establish more general rules and guidelines.
Other future work includes devising other methods of estimating user reliability than a calibration quiz. In addition, we plan to validate our simulation model by letting human users play the game under various conditions.

Disclosure statement
No potential conflict of interest was reported by the authors.

Funding
This work was partially supported by JSPS KAKENHI Grant Number 15K00324.

Notes on contributors
Daiki Kurita received his B.Eng. degree from Ritsumeikan University in 2016. He is currently a master's course student at Ritsumeikan University. His research interests include gamification and knowledge-based systems.

Boonsita Roengsamut received her B.Sc. degree from Thammasat University in 2013, and M.Eng. from Ritsumeikan University in 2015. She was engaged in research on a gamified crowdsourcing system while she was a graduate student.

Kazuhiro Kuwabara received his B.Eng., M.Eng., and Dr.Eng. degrees from the University of Tokyo, in 1982, 1984, and 1997 respectively. In 1984, he joined Nippon Telegraph and Telephone Public Corporation (NTT), and was engaged in research and development on knowledge-based systems, multi-agent systems and socialware. He was with ATR Intelligent Robotics and Communication Laboratories in during 2003–2006. Currently he is a professor at the College of Information Science and Engineering, Ritsumeikan University.

Hung-Hsuan Huang is an associate professor at the College of Information Science & Engineering, Ritsumeikan University since 2013. He received his B.S. and M.E. from National Cheng-Chi University and National Taiwan University in 1998 and 2000, respectively. In 2009, he received his PhD from the Graduate School of Informatics, Kyoto University, Japan. He then worked as a postdoctoral researcher at the Department of Computer and Information Science, Seikei University until he shifted to Ritsumeikan University as an assistant professor in 2010. His research interests are on applied artificial intelligence and human-computer interaction fields. Currently, he intensively focuses his works on embodied conversational agents (ECAs) and multi-modal interaction. He is the members of ACM, JSIAI, IPSJ, HIS, IEICE, VRSJ, and TAAI.

References
Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data – the story so far. International Journal on Semantic Web and Information Systems, 5(3), 1–22.
Cedeño, J. P., & Candan, K. S. (2011). R2DF framework for ranked path queries over weighted RDF graphs. In R. Akerkar (Ed.), Proceedings of the international conference on web intelligence, mining and semantics (pp. 40:1–40:12). New York, NY: ACM.
Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining “gamification”. In A. Lugmayr, H. Franssila, C. Safran, & I. Hammouda (Eds.), Proceedings of the 15th international academic mindtrek conference: Envisioning future media environments (pp. 9–15). New York, NY: ACM.
Doan, A., Ramakrishnan, R., & Halevy, A. Y. (2011). Crowdsourcing systems on the world-wide web. Communications of the ACM, 54(4), 86–96.
Feyisetan, O., Simperl, E., Van Kleek, M., & Shadbolt, N. (2015). Improving paid microtasks through gamification and adaptive furtherance incentives. In Proceedings of the 24th international conference on world wide web (pp. 333–343). Republic and Canton of Geneva: International World Wide Web Conferences Steering Committee.

Howe, J. (2009). Crowdsourcing: Why the power of the crowd is driving the future of business. New York, NY: Crown Business.

Ipeirotis, P. G., & Gabrilovich, E. (2014). Quizz: Targeted crowdsourcing with a billion (potential) users. In Proceedings of the 23rd international conference on world wide web (pp. 143–154). New York, NY: ACM.

Jovanovic, M. (2015). Gamifying knowledge maintenance. Retrieved from https://www.essence-network.com/wp-content/uploads/2015/08/GamifyingKnowledgeMaintenance_Jovanovic.pdf.

Jućzyn, M., Adamska, P., Rafalak, M., Balcerzak, B., Kąkol, M., & Wierzbicki, A. (2014). Threats of using gamification for motivating web page quality evaluation. In K. Marasek, & M. Sikorski (Eds.), Proceedings of the 2014 multimedia, interaction, design and innovation international conference on multimedia, interaction, design and innovation (pp. 14:1–14:8). New York, NY: ACM.

Kobren, A., Tan, C. H., Ipeirotis, P., & Gabrilovich, E. (2015). Getting more for less: Optimized crowdsourcing with dynamic tasks and goals. In Proceedings of the 24th international conference on world wide web (pp. 592–602). Republic and Canton of Geneva: International World Wide Web Conferences Steering Committee.

Kurita, D., Roengsamut, B., Kuwabara, K., & Huang, H.-H. (2016). Knowledge base refinement with gamified crowdsourcing. In T. N. Nguyen, B. Trawiński, H. Fujita, T.-P. Hong (Eds.), Intelligent information and database systems: 8th Asian conference, ACIIDS 2016, Da Nang, Vietnam, March 14–16, 2016, Proceedings, Part I (Vol. 9621, pp. 33–42). Berlin : Springer.

Lieberman, H., Smith, D., & Teeters, A. (2007). Common consensus: A web-based game for collecting commonsense goals. In ACM workshop on common sense for intelligent interfaces, Honolulu, Hawaii, USA.

Morschheuser, B., Hamari, J., & Koivisto, J. (2016). Gamification in crowdsourcing: A review. In 49th Hawaii international conference on system sciences (HICSS), Koloa, HI, USA, Jan 5–8, 2016 (pp. 4375–4384).

Rodosthenous, C., & Michael, L. (2016). A hybrid approach to commonsense knowledge acquisition. In D. Pearce, & H. Sofia Pinto (Eds.), Proceedings of the 8th European starting AI researcher symposium (STAIRS 2016) (pp. 111–122). Amsterdam: IOS Press.

Roengsamut, B., Kuwabara, K., & Huang, H.-H. (2015). Toward gamification of knowledge base construction. In Proceedings of the 2015 international symposium on innovations in intelligent systems and applications (INISTA 2015), Madrid, Spain (pp. 20–26).

Ryan, R. M., & Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. American Psychologist, 55(1), 68–78.

Saito, Y., Roengsamut, B., & Kuwabara, K. (2014). Incremental refinement of linked data: Ontology-based approach. In N. Nguyen, B. Attachoo, B. Trawiński, & K. Somboonviwat (Eds.), Intelligent information and database systems. ACIIDS 2014 (Vol. 8397, pp. 133–142). Cham: Springer International Publishing.

Šimko, J., & Bieliková, M. (2014). Semantic acquisition games – harnessing manpower for creating semantics. Cham: Springer International Publishing. (978-3-319-06115-3).

Siorpaes, K., & Hepp, M. (2008). Games with a purpose for the semantic web. IEEE Intelligent Systems, 23(3), 50–60.

Thaler, S., Simperl, E., & Siorpaes, K. (2011). Spotthelink: Playful alignment of ontologies. In D. Shin (Ed.), Proceedings of the 2011 ACM symposium on applied computing (pp. 1711–1712). New York, NY: ACM.

von Ahn, L., & Dabbish, L. (2004, April). Labeling images with a computer game. In M. Y. Vardi (Ed.), Proceedings of the SIGCHI conference on human factors in computing systems (pp. 319–326). New York, NY: ACM.

von Ahn, L., & Dabbish, L. (2008). Designing games with a purpose. Communications of the ACM, 51(8), 57–67.