Regular Paper

Additional Operations of Simple HITs on Microtask Crowdsourcing for Worker Quality Prediction

Yu Suzuki[1,a)] Yoshitaka Matsuda[1] Satoshi Nakamura[1]

Received: April 16, 2018, Accepted: October 2, 2018

Abstract: In microtask crowdsourcing, low-quality workers damage the quality of work results. Therefore, if a system automatically eliminates the low-quality workers, the requesters will obtain high-quality work results with low wages. When we consider simple Human Intelligent Tasks (*HIT*s), such as yes-no questions of a labeling task, the requesters have difficulty assessing the worker quality only from the work results. Therefore, we need a method to accurately predict the worker quality automatically from the behaviors of workers, such as working time and the number of clicks. When we accurately predict the worker quality, we are able to prepare many features from the worker behaviors. However, when we submit simple HITs, we can capture only a small number of behaviors of workers, then the accuracy of predicted worker quality will be low. To solve this issue, we propose a method to insert into the simple task of obtaining many features of worker behaviors. We prepared a classification task of tweets as simple HITs. We added a button to the work screen. The workers can browse the target tweets on the work screen during the time the workers are pressing the button, but the workers cannot browse the target tweets when the workers have released the button. Using this button, we can obtain six more kinds of features of worker behaviors. Using our method, we can improve the recall ratio 12% of identifying low-quality workers. However, as the load of workers increases, then the processing time becomes longer, and the motivation of workers decreases. From this result, we also discovered that there is a trade-off between the number of obtained behaviors and the load of workers.

Keywords: crowdsourcing, worker quality, machine learning, random forest, microtask, active intervention

1. Introduction

In microtask crowdsourcing, not all workers are able to produce results of the required quality. Moreover, some workers may have malicious intent and insufficient knowledge about the work to be done [1]. In this paper, *worker quality* is quantified as the probability of a worker doing the work correctly. This definition is followed by Refs. [2] and [3]. By hiring high-quality workers and by eliminating or instructing low-quality workers, requesters can reduce the number of inaccurate work results. Accordingly, assessments of worker quality should be viewed as essential for improving the quality of task results.

Here, we consider simple human intelligence tasks (HITs), such as answering *yes-no* questions as annotations. In such HITs, requesters find it difficult to determine the worker quality only from their results. This is because the workers who correctly select yes or no without browsing their work screens cannot be distinguished from those who carefully select the options. Moreover, requesters cannot prepare correct answers if they have the intention that only the workers should decide the results and the requesters do not want to bias the results.

To solve this problem, researchers have proposed several methods that measure worker behaviors, such as working time and number of clicks, for predicting their quality [4], [5], [6], [7]. The systems embodying these approaches automatically obtain worker behaviors on their work screens and estimate worker quality by using machine learning. The research has mainly focused on methods to analyze behaviors, rather than methods to analyze the behaviors of workers. Moreover, the target tasks in the research tend to be relatively complex, because the systems should gather many features from the workers. Therefore, these methods are difficult to apply to relatively simple HITs for which the system can obtain only a small number of behaviors from the work screens. To predict worker quality accurately, we should extract as many features from the behaviors as possible.

If we cannot capture many features from the behaviors during simple tasks, the accuracy of the predicted worker quality would be low. To solve this problem, we propose a method that adds operations on the work screens for the purpose of obtaining many features from their behaviors. We prepared a classification task of tweets, which we consider to be a simple task, as a baseline task. The purpose of this task is to classify which tweets are related to Kyoto sightseeing. We added a button to the work screen, which we call the “proposed task.” In the proposed task, workers can browse the target tweets only while they are pressing the button. Whereas we can obtain five kinds of behavior from the baseline task, and six more kinds of behavior from the proposed task.

In the proposed task, the recall ratio of low-quality workers improved by at least 12% from the baseline task. We also found that behaviors, such as browsing time and browsing count, obtainable only with the proposed task play an important role in predicting the worker quality. On the other hand, we noticed a trade-off relationship between processing time and worker dropout rate. The findings in this research can be used as guidelines for designing

[1] Nara Institute of Science and Technology, Ikoma, Nara 630–0192, Japan
[a)] ysuzuki@is.naist.jp

© 2019 Information Processing Society of Japan

crowdsourcing systems that strike a balance between prediction accuracy of quality and workload of workers.

We survey related work in Section 2. We describe the baseline HIT in Section 3. Then, we describe how we added operations on the work screens to obtain more features in Section 4. We describe our empirical experiments in Section 5 and conclusions in Section 6.

2. Related Work

There are three major approaches to predicting worker quality: 1) gold tasks, 2) analysis of work results, and 3) assessment of workers by analyzing their behavior.

The gold task approach mixes HITs that have answers (these are prepared by the requesters) and HITs which do not have answers. By comparing the answers of the workers with those of the requesters, one can automatically calculate the worker quality [8]. For individual tasks, it is necessary to create a gold task that correctly judges the quality of the worker, the result of which is added labor and costs for the requester. In addition, it is necessary to pay the worker for performing the gold task.

The approach predicts the worker quality from their results [9], [10]. Redundancy-based methods assign multiple workers to each HIT and aggregate their results. These techniques are used [11] for obtaining high-quality work results. For example, in a text labeling task, multiple workers are requested to add a label to the same piece of text. Then, the system integrates the obtained labels by majority vote [12], [13] or by using an expectation-maximization (EM) algorithm [9], [14]. The requester does not need to prepare correct answers when using this method. The basic idea of the redundancy-based method is that the majority of the labels are the correct answers. Therefore, it does not work well when low-quality workers are the majority. Also, for requesters, the wages and temporal costs increase in proportion to the number of workers made redundant. By contrast, our method reduces the number of workers assigned to each HIT. Moreover, it can be combined with the redundancy-based method to output high-quality work results at a lower cost.

Approaches that predict worker quality from their behavior have been proposed [5], [6], [7], [15]. They do not require the creation of a gold task and do not assume redundancy; hence, they make it possible to reduce costs. We decided to focus on this sort of approach in our research.

Rzeszotarski et al. [5] proposed a method that uses worker behavior for predicting the worker quality. They gathered feature data such as the number of clicks, keyboard operations, and processing time from the workers. On the other hand, Hirth et al. [6] used the following features: the duration in which the worker reads the target text, and the answer time in which the answer is considered as determined from scrolling of the page or the click interval of the radio button. Moreover, they predicted the quality of the workers by using a machine learning algorithm that took the workers' behaviors as input. They considered work events such as mouse and keyboard operations to be worker behaviors.

Mok et al. [15] tackled the problem of extracting worker behavior from measurements in detail. They analyzed the mouse cursor position by using submovement analysis [16] and grasped the behavior of the worker not by the timing of the event but by the flow. In addition, machine learning algorithms have been used for estimating the worker quality as well. Rzeszotarski et al. [7] assumed that preparing data on fully trained workers in advance improves the accuracy prediction for the worker quality. They used a machine learning model for predicting the quality of worker, where the behavior of fully trained workers was used as training data.

In these studies, the target HITs were relatively complicated tasks from which the systems could identify many kinds of behavior. For example, Rzeszotarski et al. imposed a word classification task; this single HIT can be divided into multiple HITs, and high-quality results can be obtained by dividing the task as finely as possible [1], [17].

On the other hand, there are many simple HITs, such as text labeling tasks and image annotation tasks, submitted by many requesters. However, if the number of features the system obtains is limited, the above methods do not work so well and cannot be applied as is. On simple HITs, workers do not move the mouse cursor and do not scroll page frequently [18]. Therefore, we cannot acquire enough behavior data for the machine learning model to accurately predict the worker quality. Furthermore, with only a few kinds of behavior, it is difficult to judge the worker quality even manually. As a solution to these problems, in this research, we extracted more features from simple HITs with little expression of behavior by adding more operations to the worker screens.

3. Baseline

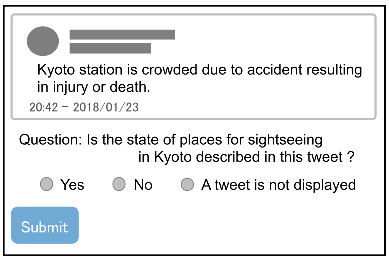
We describe the baseline task, in particular, the procedure followed by the workers and the method to predict worker quality in the baseline task. We selected the features that we used in this baseline method by referring to Rzeszotarski et al. [7].

3.1 HITs

We devised a task for labeling twitter texts. Here, the requester needs to find texts which are relevant to sightseeing in Kyoto. We prepared a set of tweets of which some were relevant to sightseeing in Kyoto. Then we had workers browse several tweets and classify whether they are useful for tourists visiting Kyoto by following both of two evaluation criteria:

- Tourists can (partially) understand the (current or past) situation of Kyoto.
- The situation is useful for sightseeing in Kyoto.

The workers were to browse a tweet, consider whether it satisfies these two criteria, and select "Yes," "No," or "Tweet is not displayed" for each tweet. **Figure 1** shows a work screen for this baseline task. For example, if there is a tweet "Kyoto station is



Kyoto station is crowded due to accident resulting in injury or death.

20:42 – 2018/01/23

Question: Is the state of places for sightseeing in Kyoto described in this tweet ?

○ Yes ○ No ○ A tweet is not displayed

Fig. 1 Work screen for the baseline task.

© 2019 Information Processing Society of Japan

Table 1 Worker behaviors used for baseline method and proposed method ($B_1, B_2, \cdots, B_5$).

ID	Behavior	Explanation	Sample
B_1	Processing time	Total time from the display of the work page until pressing the send button (seconds)	5, 10, 60
B_2	Processing time per character	Processing time divided by the number of characters of tweet (seconds)	0.05, 0.1, 1
B_3	Response time	Time from display of work page to selection of answer (seconds)	5, 10, 60
B_4	Response time per character	Response time divided by the number of characters of tweet (seconds)	0.05, 0.1, 1
B_5	Number of responses	Number of changes in answer (times)	1, 2, 5

very crowded" on the work screen, workers should select "Yes," because when the tourists browse this tweet, they can avoid going to Kyoto station when it is crowded. On the other hand, if there is a tweet "I am in Kyoto," the workers should select "No," because the tourist does not learn any useful information from this tweet. The option "Tweet is not displayed" is used if someone has deleted the tweet. Our crowdsourcing platform cannot display tweets on the work screen if they have been deleted from the original twitter site. When a worker pushes the submit button, the next tweet is displayed.

This task is easy to answer for a human, but difficult for machine learning. For example, if there is a tweet "*Kyoto station is crowded.*," it should be easy for both machine learning and a human to select the appropriate option "Yes," because there are many terms which are related to sightseeing in Kyoto, such as Kyoto, station, and crowded. However, if there is a tweet "*This train is crowded. I would have liked to go sightseeing in Kyoto unless it is crowded.*," it is difficult to know whether this text is useful or not, because there is no information about Kyoto. When this tweet is processed automatically, the system may select "Yes," because there are many keywords related to Kyoto sightseeing in the text, such as Kyoto, train, and crowded.

3.2 Worker Quality Prediction

Here, we describe a method for predicting the worker quality from the worker behaviors on the work screen. The extracted features used in the baseline method are almost the same as those used by Rzeszotarski et al. [7]. We used supervised learning for estimating the worker quality. We assessed the workers who processed at least 100 HITs.

3.2.1 Data

3.2.1.1 Feature Vectors as an Input of the Classifier

First, we describe the features of the worker behaviors. We monitored the work screens and obtained five kinds of worker behaviors $B_1, B_2. \cdots, B_5$ (**Table 1**). We acquired these behaviors every time a worker processed one HIT. Then, we calculated the average value, median value, maximum value, minimum value, standard deviation, and entropy of each behavior obtained for each worker and used them as feature values for each worker. The feature values of each behavior $\boldsymbol{x_w^i}$ $(i = 1, 2, \cdots, 5)$ which correspond to $B_1, B_2, \cdots, B_5$ are defined as follows:

$$\boldsymbol{x_w^i} = [\ B_i^{Ave}\ B_i^{Med}\ B_i^{Min}\ B_i^{Max}\ B_i^{Std}\ B_i^{Ent}\ ], \tag{1}$$

where B_i^{Ave} is the average value, B_i^{Med} is the median value, B_i^{Min} is the minimum value, B_i^{Max} is the maximum value of B_i B_i^{Std} is the standard deviation, and B_i^{Ent} is the entropy of B_i.

In addition, we used the number of processed HITs N_w and the total processing time T_w as the features of worker behaviors. Accordingly, the feature vector $\boldsymbol{x_w}$ of worker w is defined as follows:

$$\boldsymbol{x_w} = \text{concat}[\ N_w\ T_w\ \boldsymbol{x_w^1}\ \boldsymbol{x_w^2}\ \boldsymbol{x_w^3}\ \boldsymbol{x_w^4}\ \boldsymbol{x_w^5}\ ] \tag{2}$$

3.2.1.2 Worker Quality as an Output of the Classifier

The values which the classifier predicts are worker qualities. We used the correct answer rate r_w as follows:

$$r_w = \frac{M_w}{N_w} \tag{3}$$

where M_w is the number of processed HITs with correct answers. We created the correct answers using majority voting. We assigned ten workers to each tweet and set the correct option as the one selected by six or more workers. We checked the data and found that there is no tweet which the maximum number of votes was not less than half of the total number of votes. Moreover, there are less than 1% of tweets in which the maximum number of votes were six. Therefore, we believe that this dataset is adequate to be used.

Next, we calculated the worker quality Q_w to determine which workers were low-quality and which were high-quality as follows:

$$Q_w = \begin{cases} -1 & \text{if } r_w \leq \beta \\ 1 & \text{else} \end{cases} \tag{4}$$

where $Q_w = -1$ means that the quality of w is low, and $Q_w = 1$ means that the quality of w is high. α is the ratio of the low-quality workers to the total number of workers. β is a threshold of low-quality workers. We treat a worker as low-quality if his/her correct answer rate is less than β. For example, if α equals 0.1, 10% of the workers are low-quality and $Q_w = -1$ and Q_w of the other workers equals 1. If the highest correct answer rate of the low-quality workers is 0.7, β is 0.7. If α is given, β is automatically calculated.

In our experiments, almost all workers were high-quality. Therefore the numbers of high-quality and low-quality workers were imbalanced in the training data. When we classified workers, almost all of them were classified into major classes incorrectly. Therefore, we used the SMOTE algorithm [19] which virtually increases the samples of a small number of classes so that the numbers of high-quality and low-quality workers would almost be the same.

3.2.2 Classifier Construction

The classifier f was as follows:

$$f : \boldsymbol{x_w} \mapsto \hat{Q}_w \in \{-1, 1\} \tag{5}$$

where $\hat{Q}_w$ is the quality of worker w predicted by the classifier f. To construct f, we prepared a set of workers with x_w and Q_w. We used random forests [20] because the classifier should be resistant to outliers. There are many outliers in the input feature values.

Table 2 Hyperparameters.

name	candidates
# estimater	2, 3, 4, 5, 6, 7, 8, 9, 10
maximum # of features	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ‘auto’, None

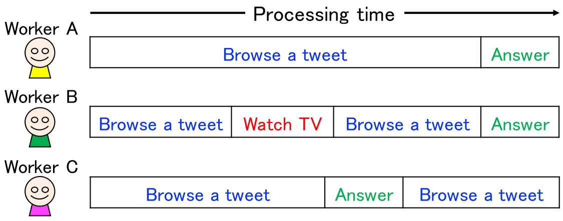


Fig. 2 Example of processing times of three workers.

For example, suppose a worker performs an HIT while watching TV. If he or she becomes very distracted, the maximum processing time per HIT could be more than one hour. Friedman et al. [21] discovered that the random forest is resistant to this kind of outlier. They also discovered that random forests could accurately predict if many features were not useful for prediction. We used a grid search for setting appropriate hyperparameters in the candidates shown in **Table 2**.

4. Proposed Method

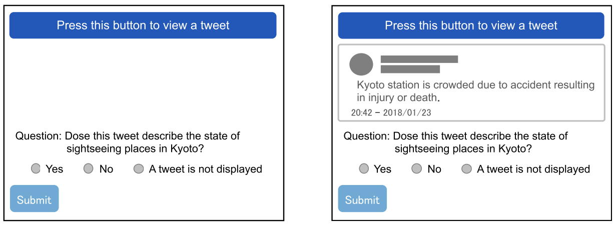
Here, we describe the procedure of the HITs, the method to predict worker quality, and the differences between the baseline and proposed method.

4.1 Basic Idea

First, let us consider the workflow in the tweet classification task described in Section 3.1 and discuss the features related to processing time. **Figure 2** shows several examples of worker behaviors in the HITs. The figure shows three types of workers; *A*, *B*, and *C*. Worker *A* first browses a tweet for a long time before answering. Worker *B* works while watching TV; she does not continuously browse tweets. Worker *C* first browses a tweet and answers; then she browses the tweet again to re-check the answer. From these worker behaviors, we intuitively find that *C* is the highest quality and *B* is the lowest quality.

Current crowdsourcing platforms do not monitor worker behaviors directly; the platform could not understand whether the workers were browsing tweets, watching TV, or doing something else. In this example, the processing times of *browsing a tweet* and *answering* for each worker are actually different, but the system treats these times as the same. Therefore, if we extract features listed at the baseline method in Table 1 from the behaviors of the three workers, the feature values would be the same; the classifier considers the three workers as having the same quality.

Based on the above, we believe that we can estimate the worker quality more accurately if we could obtain more data on their behavior. For this reason, the proposed method obtains the behavior of the browsing part of the task. In particular, we added operations on the work screens to obtain the browsing time and the number of browsings. We verified the effectiveness of such behaviors which have not been dealt with in previous research on predicting the worker quality.



Question: Dose this tweet describe the state of sightseeing places in Kyoto?

Kyoto station is crowded due to accident resulting in injury or death.

Question: Dose this tweet describe the state of sightseeing places in Kyoto?

(a) Unpressed (b) Pressed

Fig. 3 Work screen of intervening task.

4.2 HITs

In the proposed method, tweets are not simply displayed on the work screen. Instead, a button, which we call the *browsing button*, has to be pressed and held down to display the tweets. We show the work screen in **Fig. 3**. The left side of this figure is the work screen when the browsing button is not pressed, and the right side is the work screen when the worker presses the browsing button. When the worker releases the browsing button, the tweet disappears from the work screen, as indicated on the left side of the figure.

The system can obtain the browsing time and frequency from the browsing button operations. For example, if a worker is doing a task while watching TV, the processing time B_1 will likely be longer.

4.3 Feature Values

Using the browsing button described in Section 4.2, we constructed the feature values shown in Table 1 and **Table 3**. We predicted the worker quality using random forests, as in the baseline method. The feature vectors $\boldsymbol{x}_w^p$ of worker w in the proposed method are defined as follows:

$$\boldsymbol{x}_w^p = \mathrm{concat}[\; N_w \;\; T_w \;\; \boldsymbol{x}_w^1 \;\; \boldsymbol{x}_w^2 \;\; \dots \;\; \boldsymbol{x}_w^{11} \;] \qquad (6)$$

where x_w^i $(i = 1, 2, \cdots, 11)$ is the feature value which corresponds to $B_1, B_2, \cdots, B_{11}$ defined by Eq. (1).

5. Experimental Evaluation

In this section, we describe our experimental evaluation to verify whether the features of the worker behaviors obtained by the operations that were added to the work screens are effective for predicting worker quality. We also investigate the effect of the browsing buttons on the relationship between the prediction accuracy of worker quality and workload of workers.

5.1 Experimental Setup

We constructed our crowdsourcing platform as a Web application using Ruby on Rails 5.1 and Oracle Database 12*c*. All workers we hired performed the HITs on our platform. We used jQuery [*1], a javascript library, to obtain the worker behaviors on the work screens.

5.1.1 Worker

We hired workers through Crowdworks [*2], a major crowdsourcing platform in Japan. All workers who applied for our task were hired without filtering; we did not select the workers to be

[*1] https://jquery.com/

[*2] https://www.crowdworks.jp/

© 2019 Information Processing Society of Japan

Table 3 Worker behaviors used only in proposed method ($B_6, B_7, \cdots, B_{11}$).

ID	Behavior	Explanation	Sample
B_6	Browsing count	Number of times tweet display button was pressed (times)	1, 2, 5
B_7	Browsing time	Total time while holding down tweet display button (seconds)	5, 10, 60
B_8	Browsing time per character	Browsing time divided by the number of characters of tweet (seconds)	0.05, 0.1, 1
B_9	Browsing time per count	Browsing time divided by the browsing count (seconds)	5, 10, 60
B_{10}	Browsing time per count and character	B_7 divided by B_6 and B_8 (seconds)	0.05, 0.1, 1
B_{11}	Browsing ratio	Percentage of browsing time out of total processing time	0.1, 0.5, 0.8

Table 4 Number of workers, HITs, and tweets.

	# workers	# active workers	# early departed workers	# processed HITs	# tweets
Baseline	439	250	189	132,594	–
Proposed	486	255	231	108,836	–
Sum	793	439	354	241,430	26,713

hired in advance. Workers can earn 30 JPY (about 0.3 USD) as wages for classifying 100 tweets. The average worker classified about 450 tweets per hour and earned 135 JPY, which is almost equal to the standard crowdsourcing wage (1.38 USD/hour) [22]. Workers could freely suspend or resume their work. As a result, some workers processed fewer than 100 tweets, while others processed more than 10,000 tweets.

5.1.2 Data

Table 4 shows the number of workers we hired and the total number of HITs when using the proposed method and the baseline method. Here, "*#workers*" refers to the total number of workers who applied for our task and engaged in it more than once, and "*#active workers*" refers to the workers who processed more than 100 HITs. "*#early departed workers*" refers to the workers who are not active workers. "*#processed HITs*" is the number of tasks processed by all workers. "*#tweets*" is the number of tweets classified by more than one worker. In our system, we assigned ten workers to each tweet. However, more than two workers reported that the tweets had been deleted, so we canceled the assignments to those tweets.

5.2 Worker Quality Prediction

Here, we compare the worker qualities predicted by the following three methods: the baseline method, the proposed method, and the proposed method with limited features (proposed-LF). The features of the baseline method are almost the same as those described in Rzeszotarski's paper [7]. Moreover, those of the proposed method with limited features, which we call *proposed-LF*, used only the features of the baseline system. Thus, in the proposed-LF, the prediction accuracy of the worker quality is the same as in the baseline method, while the workload of the workers is the same as in the proposed method.

Furthermore, we discuss the effectiveness of the behaviors obtained using the browsing button. We also analyze the properties of the correctly predicted workers and those of the workers who were not correctly predicted.

5.2.1 Evaluation of Prediction Accuracy

We evaluated the classifiers by five-fold cross-validation. We divided the workers into five groups; four groups were used for learning data, and the remaining group was used for calculating precisions and recalls. We repeated this process five times.

Table 5 Correlation between α and β.

α	Method	β
0.05	Baseline	65.5%
	Proposed	62.8%
0.1	Baseline	73.0%
	Proposed	72.6%
0.15	Baseline	80.7%
	Proposed	78.7%

Table 6 Clustering results.

α	Method	Precision	Recall	F-value
0.05	Baseline	0.17	0.55	0.26
	Proposed	**0.21**	**0.75**	**0.33**
	Proposed-LF	0.16	0.58	0.25
0.1	Baseline	0.21	0.72	0.32
	Proposed	**0.23**	**0.84**	**0.36**
	Proposed-LF	0.20	0.64	0.30
0.15	Baseline	0.22	0.62	0.33
	Proposed	**0.32**	**0.79**	**0.45**
	Proposed-LF	0.27	0.71	0.39

Table 5 shows the correlation between α and β, and **Table 6** shows the clustering result for each value of α. From this table, we conclude that the accuracy of the proposed method is better than that of the baseline method and that of the proposed-LF method.

We do not compare the value of α and β of the proposed method and that of the baseline method. This is because the workers of our proposed method and that of the baseline method are not the same, we cannot compare the worker quality of each worker. Instead of comparing the baseline system with the proposed system, we compared the proposed method with the proposed-LF system, which simulates the baseline system with the workers of the proposed method.

We do not set the value of α to a lower value than 0.05, because the number of workers is too few. When the values of α are set to 0.01, the number of low-quality workers is 2. If a classifier can classify these workers as low quality, the precision ratio of detecting low-quality workers should be 100%, but we think that this precision ratio is meaningless. We think that at least 10 workers should exist to obtain reliable results. Therefore, we assume that the lower bound of α should be 0.05.

Also, we do not set the value of α to a higher value than 0.15. If we set the value of α to 0.15, the workers whose correct answer rates are 78% are considered as low-quality workers. Intuitively, the workers who select appropriate options with 80% accuracy should be considered as high-quality workers. Therefore, we did

© 2019 Information Processing Society of Japan

not set the value of α or β to a higher value.

Here we consider the case where α is 0.1 as an example; this means that the workers with correct answer rates of lower than 10% are considered low quality, and the other workers are considered high quality. The correct answer rate β, which is the boundary between the low-quality workers and the high-quality workers, was 73.0% in the baseline method and 72.6% in the proposed method.

The confusion matrix of the baseline method, proposed method, and proposed-LF are shown in **Table 7**. It is more useful for requesters to identify low-quality workers accurately than to identify high-quality workers, because they can then eliminate or train low-quality workers who would otherwise degrade the quality of the work results. Therefore, the recall ratio of low-quality workers is important. The recall ratios of low-quality workers were 0.72, 0.84, and 0.64 for the baseline, proposed, and proposed-LF methods, respectively. From these results, we discovered that the proposed method has the highest accuracy of the three at predicting low-quality workers. However, it also predicted many high-quality workers to be low-quality. Therefore, there is room for improving it.

5.2.2 Effective Behaviors

The five-fold cross-validation calculated the importance degrees for each of five classifications. **Figure 4** shows the top-5 important features in the proposed method. From these figures, we discovered the features obtained using the browsing button $(B_6, B_7, \cdots, B_{11})$ were more important than those using the baseline method $(B_1, B_2, \cdots, B_5)$.

From Fig. 4, we discovered that the minimum browsing time per count (B_9-minimum) is selected as an important behavior at every threshold α. **Figure 5** shows the relationship between B_9-minimum and the correct answer rate. Each point represents a worker. The workers with a small B_9-minimum have a high correct answer rate. Therefore, we can say that a worker has a high correct answer rate if B_9-minimum is high. However, the con-

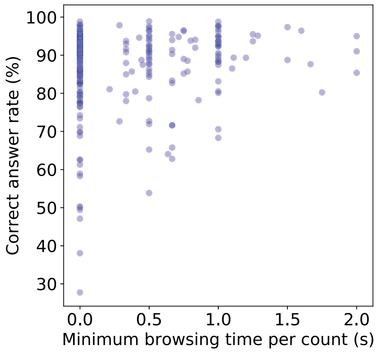


Fig. 5 B_9-minimum vs. correct answer rate.

Table 7 Confusion matrix (L: low-quality, H: high-quality).

(a) Baseline

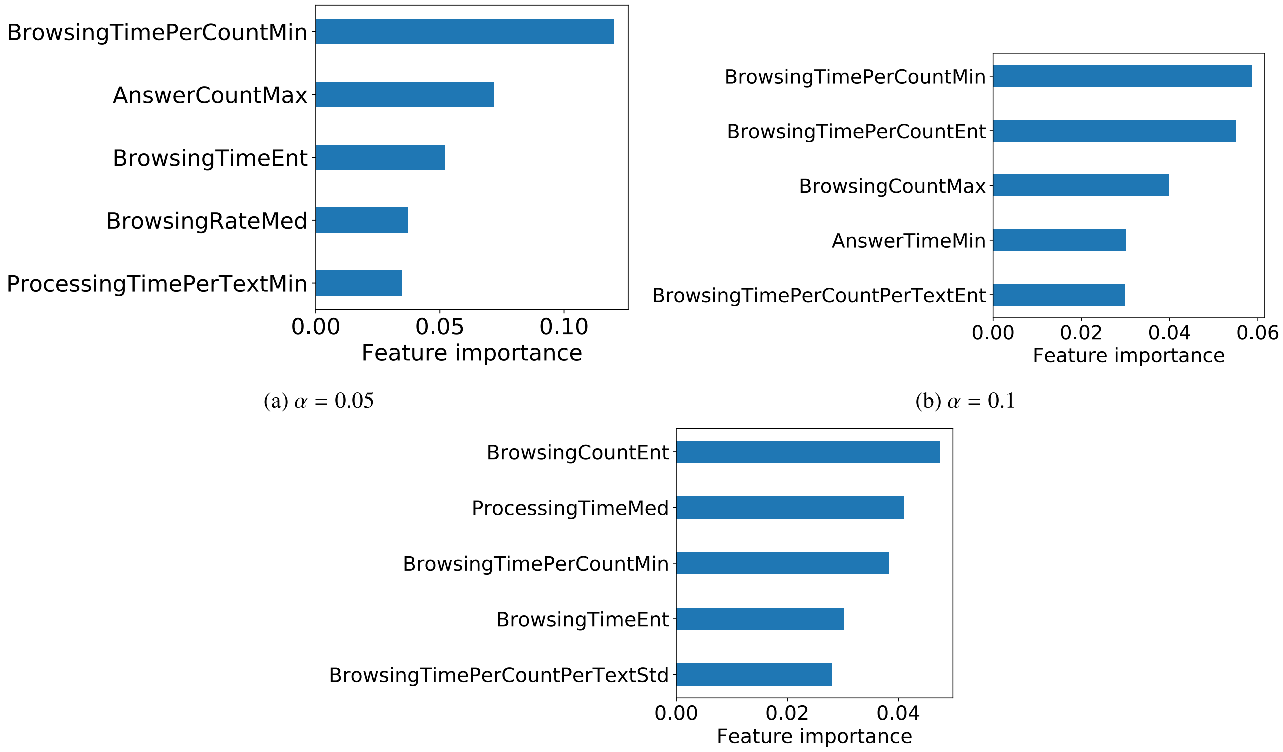
		Predict	
		L	H
Collect	L	18	7
	H	69	156

(b) Proposed

		Predict	
		L	H
Collect	L	21	4
	H	72	158

(c) Proposed-LF

		Predict	
		L	H
Collect	L	16	9
	H	65	165



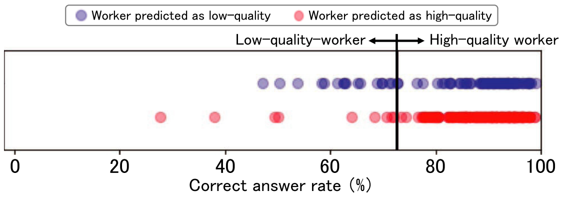
(a) α = 0.05

(b) α = 0.1

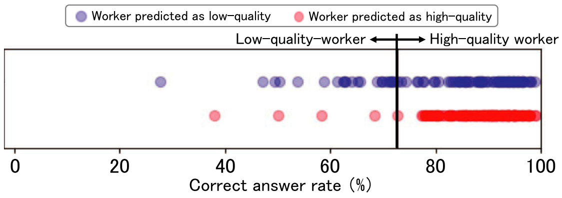
(c) α = 0.15

Fig. 4 Top-5 important features.

© 2019 Information Processing Society of Japan



(a) Proposed method-LF



(b) Proposed method

Fig. 6 Prediction results ($\alpha = 0.1$).

verse is not necessarily true; we cannot say that a worker has a low correct answer rate if his/her B_9-minimum is low. Therefore, although the proposed method can detect low-quality workers, it may also misclassify high-quality workers as low-quality ones. This fact also appears in the confusion matrix of Table 7.

Also, from Fig. 4, we discovered that the features about the browsing count (B_6) are important for classification. In the worker behavior records, many high-quality workers browsed the tweets many times, and some browsed even after selecting options.

5.2.3 Discussion of Worker Behaviors

We compared the proposed and the proposed-LF method to discover which features are effective for classification of the workers. The workers in the proposed method and the proposed-LF method are the same. **Figure 6** shows classification results when $\alpha = 0.1$. Figure 6 (a) is the classification result of the proposed-LF method, and Fig. 6 (b) is the classification result of the proposed method. Each point represents a worker, and the horizontal axis is the correct answer rate of the workers. In Fig. 6, the workers represented in blue are predicted as low-quality, and the workers represented in red are predicted as high-quality. In an ideal classifier, blue circles are to the left of the boundary between low-quality and high-quality workers, and red circles are on the right side. Also, we discovered that high-quality workers could classify workers with lower correct answer rates as better classifiers.

5.2.3.1 Case 1: Correctly Classified by Proposed but Misclassified by Proposed-LF

Here, we discuss the lowest-quality worker. This worker had a correct answer rate of only 28% and thus should be considered low-quality. This worker is shown at the left end in the two figures of Fig. 6. In Fig. 6 (a), this worker is represented in red; this means that the proposed-LP method incorrectly predicts this worker as high-quality. On the other hand, in Fig. 6 (b), this worker is represented in blue. The proposed method thus correctly predicts the worker as low-quality.

The average processing time of this worker was 6 seconds, shorter than that of the average worker. However, as described in

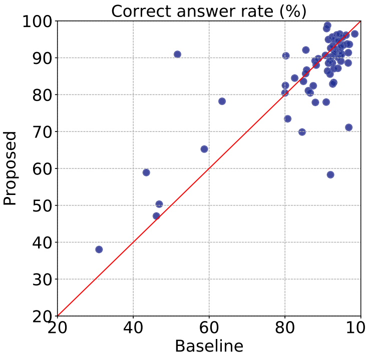


Fig. 7 Correct answer rates of workers in baseline and proposed tasks.

Section 5.2.2, workers are not always low quality if their processing time is short. As a result, the proposed-LF method misclassifies this worker as a high-quality worker. However, the worker did not browse more than 90% of the tweets, and the browsing time was at most 2 seconds, much shorter than average. Incorporating the browsing button into the task revealed this fact. Therefore, the proposed method correctly classified this worker as low-quality.

5.2.3.2 Case 2: Incorrectly Classified by Both Proposed and Proposed-LF

Here, we discuss a low-quality worker who was correctly classified as low-quality by both proposed-LF and the proposed method. The browsing count was 1.5 times more than average, and the browsing time was two times longer than average. Although the correct answer rate of this worker is low, we feel that this worker carefully read the tweets. We believe that this worker did not have enough knowledge about Kyoto or sightseeing, but worked eagerly and attentively. When we use our method, the system will misclassify this kind of worker. To solve this problem, we should develop a method to predict worker quality using not only worker behaviors but also work results and effective features obtained by other methods.

5.3 Burden on Workers of Proposed Method

In the proposed method, workers must keep pressing the button in order to read a tweet. However, while the browsing button is effective for accurately predicting worker quality, it is not directly effective in improving worker quality. It should actually decrease the usability of work screens and increase the workload of workers compared with the baseline system. In this section, we discuss the effect of the browsing button in the proposed method from three points of view: task difficulty, processing time, and motivation.

5.3.1 Task Difficulty

First, we determined whether the browsing button affects worker quality. We measured the difficulty level of the task from the differences and distributions of correct answer rates in the tasks of the proposed method and those of the baseline method. We identified workers who were hired for both the baseline and proposed tasks. Then, we calculated their correct answer rates in the proposed method and the baseline method. The results are shown in **Fig. 7**, where the horizontal axis is the correct answer rate in the baseline method, the vertical axis is the correct answer rate in the proposed method, and each point represents a worker.

© 2019 Information Processing Society of Japan

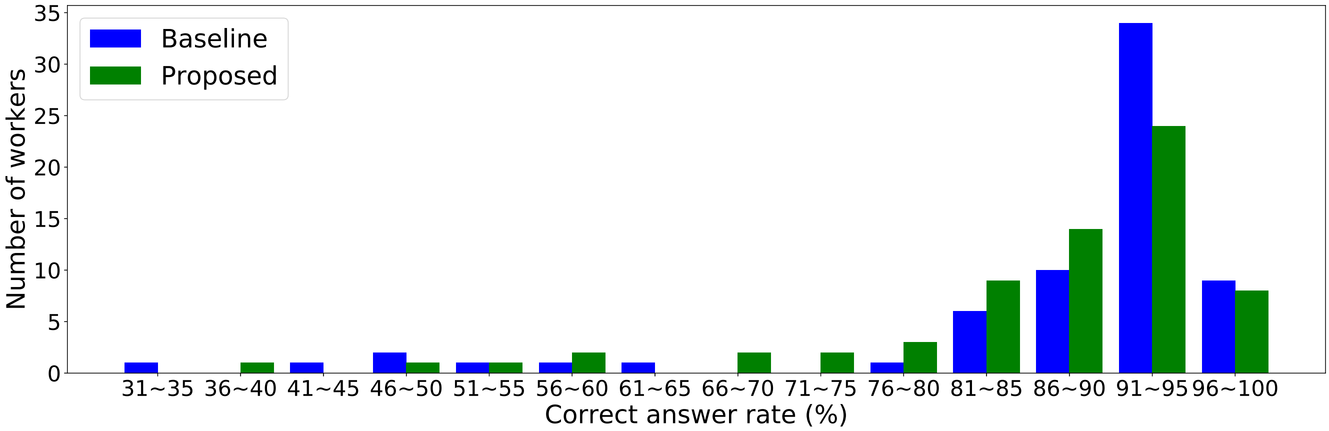


Fig. 8 Distribution of correct answer rates of workers in baseline and proposed tasks.

The correlation coefficient is 0.78; we thus know that there is a strong positive correlation between the correct answer rates for the baseline and for the proposed method.

Next, we show the distribution of correct answer rates of workers in the two tasks in **Fig. 8**. The horizontal axis is the correct answer rate, and the vertical axis is the number of workers with the correct answer rate. We used the Wilcoxon signed-rank test to determine any difference between the correct answer rates of the proposed and baseline method, but no significant difference was found ($p > 0.05$). From this figure, we can conclude that there is a possibility that the dificulties of the baseline and that of the proposed method are the same.

There is a positive correlation between the correct answer rates in the baseline method and the proposed method. Therefore, we cannot conclude that there is a difference in task difficulty, and there is a difference between the correct answer rate of the baseline method and that of the proposed method. Therefore, we can confirm that there is a possibility that the user interface does not affect worker quality.

5.3.2 Processing Time

In the proposed method, workers have to push the browsing button; they must move the mouse cursor to the browsing button each time they read a tweet. Therefore, the processing time of the proposed method should be longer than that of the baseline method.

Figure 9 shows a box plot of the distributions of processing time for the baseline and proposed method. The vertical axis of the graph is the processing time on a logarithmic scale. The median value is 6 seconds in the baseline method, and 9 seconds in the proposed method. As we expected, the proposed method increased the processing time by 3 seconds, which proved a time-consuming burden on workers. Furthermore, it took more time for all the tasks to be processed by the task requester as well.

Also, the processing times of some workers exceeded 1,000 seconds in both methods. The maximum number of characters on Twitter is currently 140, and it is hard to believe that a worker could take so much time. As mentioned in Section 4, we cannot believe that workers are always concentrating on their tasks. We imagine that these workers were processing HITs while doing other things or working on tasks across breaks.

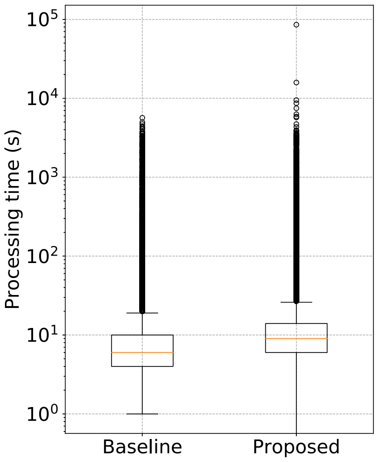


Fig. 9 Processing time.

5.3.3 Motivation

For measuring worker motivation, we use the departing ratio of HITs, which is the ratio of the number of early departed workers to the number of all workers. It is difficult to measure motivation used in a psychological research field because the target workers are an unspecified number of workers. In this experiment, we did not measure the psychological motivation. On the other hand, in this experiment, a worker can cancel the task freely at his/her convenience at any time. Therefore, assuming that the total number of workers is large, it was assumed that motivation is high. We defined the early departed workers as those who processed less than 100 HITs, because we only paid wages to those who processed more than 100 HITs. Table 4 shows the number of early departed workers. From this table, we can see that in the baseline method, 43.1% of the workers (189) departed early, while in the proposed method, 47.5% of the workers (231) departed early. The proposed method increased the departing ratio by 4.4%.

Next, **Fig. 10** shows the number of processed HITs versus the number of workers. The horizontal axis is the number of processed HITs, while the vertical axis is the number of workers. Although the wages of both the baseline and the proposed system are the same, the number of workers who process less than 20 tasks in the proposed method is larger than them in the baseline method.

Figure 11 shows the distribution of task processing times of a

© 2019 Information Processing Society of Japan

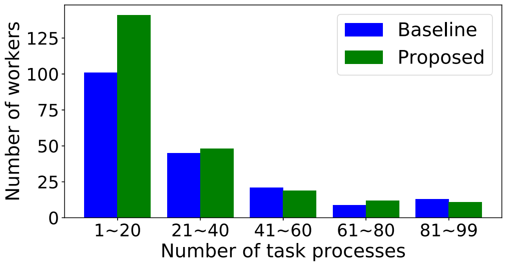


Fig. 10 Distribution of number of workers who processed less than 100 HITs.

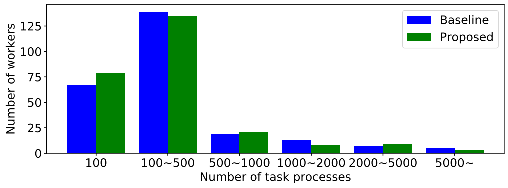


Fig. 11 Distribution of number of workers who processed at least 100 HITs.

particular worker. It shows that there is no significant difference in the distribution of task processing times between the methods. To summarize the above results, worker motivation decreased in the proposed method when the workers processed less than 100 HITs. However, the motivation of workers who processed more than 100 HITs was almost the same in the proposed and baseline method.

6. Conclusion

We proposed a method of predicting worker quality from worker behaviors in simple HITs by adding operations to the worker screens. While previous research analyzed worker behavior, we focused on obtaining more behaviors by installing a button in the task. In particular, to obtain the browsing time more accurately, the proposed method forces the worker to press a button for displaying tweets.

We performed an empirical evaluation on detecting low-quality workers in a yes-no type annotation task, which is a simple HIT. The results confirm that compared with a baseline method with no button, our method improves the recall ratio and maintains the precision ratio; this means our method can more accurately detect low-quality workers than the baseline method can. In particular, the recall rate was the highest, 0.84, when we defined a low-quality worker as one with a correct answer rate of the lower 10%. In addition, we found that behaviors measurable with the button, such as the browsing time per visit and number of viewing times, were more useful in classifying workers than those not obtained with the button. A worker with a short processing time could be either high-quality or low-quality, but by looking at a breakdown of processing time into browsing time, etc., it becomes possible to classify these workers. The disadvantage due to the installation of the buttons is that the processing time increases and the hurdles over which the worker has to go in their tasks go up.

Finally, we describe four issues as open problems. The first issue is misclassification of workers who diligently process HITs. In order not to misjudge workers who can quickly perform appropriate work and those who do not understand the goal of the task correctly, we should obtain more features not only on their behavior but also their work results.

The second issue is to verify the versatility of the proposed method. In this paper, we conducted experiments on one type of task. Although we did not consider translation tasks and article writing tasks, we believe that it is possible to apply the proposed method to tasks that give a simple evaluation by looking at certain content, for example, image labeling and questionnaires. Also, the task performed in this study was easy for many workers, with a correct answer rate of about 90%. It should also be verified for tasks in which the incorrect answer rate varies depending on the worker's ability and tasks that normally have a low overall correct answer rate.

The third issue is to consider adding operations to other tasks. The proposed method uses the browsing button to obtain the features of worker behaviors. However, this method would not be appropriate for tasks other than annotation. In future, we will consider a method to obtain more behaviors that puts less load on workers.

The fourth issue is to use the prediction of worker quality for the selected workers to earn better work results. To discover this consideration, further challenges such as how to select a worker and how much the number of worker needs to be reduced are required.

Acknowledgments The research results have been achieved by "Research and Development on Fundamental and Utilization Technologies for Social Big Data," the Commissioned Research of National Institute of Information and Communications Technology (NICT), Japan.

References

[1] Kittur, A., Chi, E.H. and Suh, B.: Crowdsourcing user studies with Mechanical Turk, *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp.453–456, ACM (2008).

[2] Oleson, D., Sorokin, A., Laughlin, G., Hester, V., Le, J. and Biewald, L.: Programmatic Gold: Targeted and Scalable Quality Assurance in Crowdsourcing, *Proc. 11th AAAI Conference on Human Computation, AAAIWS '11-11*, pp.43–48, AAAI Press (2011) (online), available from ⟨http://dl.acm.org/citation.cfm?id=2908698.2908706⟩.

[3] Vuurens, J., de Vries, A. and Eickhoff, C.: How much spam can you take? An analysis of crowdsourcing results to increase accuracy, *Proc. ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR '11)*, pp.21–26 (2011).

[4] Zhu, D. and Carterette, B.: An analysis of assessor behavior in crowdsourced preference judgments, *SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation*, pp.17–20 (2010).

[5] Rzeszotarski, J.M. and Kittur, A.: Instrumenting the crowd: Using implicit behavioral measures to predict task performance, *Proc. 24th Annual ACM Symposium on User Interface Software and Technology*, pp.13–22, ACM (2011).

[6] Hirth, M., Scheuring, S., Hoßfeld, T., Schwartz, C. and Tran-Gia, P.: Predicting result quality in crowdsourcing using application layer monitoring, *2014 IEEE 5th International Conference on Communications and Electronics (ICCE)*, pp.510–515, IEEE (2014).

[7] Kazai, G. and Zitouni, I.: Quality management in crowdsourcing using gold judges behavior, *Proc. 9th ACM International Conference on Web Search and Data Mining*, pp.267–276, ACM (2016).

[8] Kazai, G., Kamps, J., Koolen, M. and Milic-Frayling, N.: Crowdsourcing for book search evaluation: Impact of hit design on comparative system ranking, *Proc. 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.205–214, ACM (2011).

[9] Dawid, A.P. and Skene, A.M.: Maximum likelihood estimation of observer error-rates using the EM algorithm, *Applied Statistics*, pp.20–28 (1979).

[10] Raykar, V.C., Yu, S., Zhao, L.H., Jerebko, A., Florin, C., Valadez, G.H., Bogoni, L. and Moy, L.: Supervised learning from multiple experts: Whom to trust when everyone lies a bit, *Proc. 26th Annual*

© 2019 Information Processing Society of Japan

International Conference on Machine Learning, pp.889–896, ACM (2009).

[11] Smyth, P., Fayyad, U.M., Burl, M.C., Perona, P. and Baldi, P.: Inferring ground truth from subjective labelling of venus images, *Advances in Neural Information Processing Systems*, pp.1085–1092 (1995).

[12] Sheng, V.S., Provost, F. and Ipeirotis, P.G.: Get another label? Improving data quality and data mining using multiple, noisy labelers, *Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.614–622, ACM (2008).

[13] Snow, R., O'Connor, B., Jurafsky, D. and Ng, A.Y.: Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks, *Proc. Conference on Empirical Methods in Natural Language Processing*, pp.254–263, Association for Computational Linguistics (2008).

[14] Oyama, S., Baba, Y., Sakurai, Y. and Kashima, H.: Accurate Integration of Crowdsourced Labels Using Workers' Self-reported Confidence Scores, *IJCAI*, pp.2554–2560 (2013).

[15] Mok, R.K., Chang, R.K. and Li, W.: Detecting Low-Quality Workers in QoE Crowdtesting: A Worker Behavior-Based Approach, *IEEE Trans. Multimedia*, Vol.19, No.3, pp.530–543 (2017).

[16] Meyer, D.E., Abrams, R.A., Kornblum, S., Wright, C.E. and Smith, J.K.: Optimality in human motor performance: Ideal control of rapid aimed movements, *Psychological Review*, Vol.95, No.3, p.340 (1988).

[17] Alonso, O.: Implementing crowdsourcing-based relevance experimentation: An industrial perspective, *Information Retrieval*, Vol.16, No.2, pp.101–120 (2013).

[18] Ling, W., Marujo, L., Dyer, C., Black, A. and Trancoso, I.: Crowdsourcing High-Quality Parallel Data Extraction from Twitter, *ACL 2014*, p.426 (2014).

[19] Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P.: SMOTE: Synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research*, Vol.16, pp.321–357 (2002).

[20] Breiman, L.: Random forests, *Machine Learning*, Vol.45, No.1, pp.5–32 (2001).

[21] Friedman, J., Hastie, T. and Tibshirani, R.: *The elements of statistical learning*, Vol.1, Springer (2001).

[22] Horton, J.J. and Chilton, L.B.: The labor economics of paid crowdsourcing, *Proc. 11th ACM Conference on Electronic Commerce*, pp.209–218, ACM (2010).

Yu Suzuki was born in 1977. He received his M.E. and Ph.D. degrees from Nara Institute of Science and Technology in 2001 and 2004, respectively. He became an assistant professor at Ritsumeikan University in 2004, a researcher at Kyoto University in 2009, and an assistant professor at Nagoya University in 2010. He is currently an associate professor at Nara Institute of Science and Technology. His current research interests include Social Web analysis and data mining. He is a member of IPSJ, IEICE, DBSJ, IEEE-CS, and ACM.

Yoshitaka Matsuda is a graduate student of Nara Institute of Science and Technology. He received his M.E. degree from Nara Instutute of Science and Technology in 2018. His research interests are crowdsourcing, human behavior, and machine learning. He is a member of DBSJ.

Satoshi Nakamura is a Professor of Nara Institute of Science and Technology (NAIST), Japan, Project Leader of Tourism Information Analytics Team of RIKEN, Center for Advanced Intelligence Project AIP, Honorarprofessor of Karlsruhe Institute of Technology, Germany, ATR Fellow, IEEE Fellow, IPSJ Fellow. He received his Ph.D. from Kyoto University in 1992. He was Associate Professor of NAIST in 1994–2000. He was Director of ATR Spoken Language Communication Research Laboratories in 2000–2008, and Director General of Keihanna Research Laboratories, National Institute of Information and Communications Technology, Japan in 2009–2010. He received the IPSJ Yamashita Award, IPSJ Kiyasu Award and the Commendation for Science and Technology by the Minister of Education, Science and Technology, and the Commendation for Science and Technology by the Minister of Internal Affair and Communications.

© 2019 Information Processing Society of Japan