Information Sciences 512 (2020) 80–95

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Incentive mechanism for the listing item task in crowdsourcing

Shaofei Wang, Depeng Dang*

College of Information Science and Technology, Beijing Normal University, Beijing 100875, China

ARTICLE INFO

Article history:
Received 26 April 2018
Revised 8 September 2019
Accepted 25 September 2019
Available online 26 September 2019

Keywords:
Crowdsourcing
Incentive mechanism
Listing item task

ABSTRACT

Crowdsourcing is a new strategy of leveraging intelligence from a large number of workers to complete tasks. An incentive mechanism is an effective way for improving the quality of answers in crowdsourcing. However, a special but common type of crowdsourcing task, called listing item task, has not been fully investigated. In this paper, we focus on the incentive mechanism for this listing item task. In particular, we first provide a formal definition of this task. Then, we propose an effective incentive mechanism considering both the precision and recall of the answers. Next, we prove that the proposed mechanism is incentive-compatible and satisfies no free lunch criterion. Finally, we conduct a series of experiments on our crowdsourcing platform CrowdKnow and a public platform ZhiDao. The experimental results demonstrate that our incentive mechanism achieves a remarkable improvement for listing item tasks compared with other related mechanisms.

1. Introduction

Crowdsourcing is a new solution that utilizes human intelligence to complete diverse tasks that are relatively difficult for computers [14,29]. Crowdsourcing involves three important elements: requesters, workers, and platform. In particular, the requesters release their tasks with demands on the platform and set the incentive mechanism. The workers select and complete the tasks that they are good at, and they are automatically paid for their answers by the incentive mechanism [11].

Because the expertise levels of workers range widely, the quality of crowdsourcing answers provided by different workers can vary considerably [5,26]. A few studies have proved the effectiveness of incentive mechanisms for improving the quality of answers [40,42]. In general, the incentive mechanism has the following two functions to guarantee the quality of answers: 1) The workers who provide high-quality solutions should be given sufficient rewards to motivate them to complete more tasks (this function is referred to as the *incentive compatibility*, which is formally defined in Section 2.1). 2) Spammers should not be rewarded because they always submit irrelevant contents with the hope of earning a few free rewards, which pose a threat to the quality of answers (this function is recognized as the *no free lunch criterion* defined in Section 2.1).

Actually, there are many studies that focus on incentive mechanisms for various types of crowdsourcing tasks. The tasks are mainly divided into two categories: macrotasks and microtasks. Macrotasks are a few complex jobs that cannot be divided and require many hours to complete, such as program design and innovation contests [10,17]. Microtasks are simple

* Corresponding author.
E-mail address: ddepeng@bnu.edu.cn (D. Dang).

https://doi.org/10.1016/j.ins.2019.09.067
0020-0255/

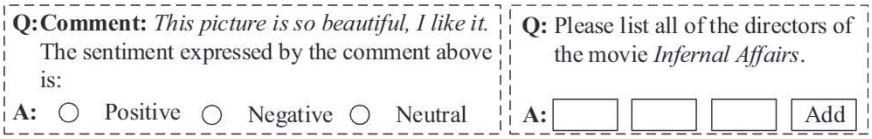

Fig. 1. Different crowdsourcing microtasks: (a) ordinary type of microtask, where the requester provides candidate options for workers to select; (b) example of a listing item task, where it is difficult for the requester to provide candidate options.

tasks that are repetitive but easy for workers to accomplish [8]. They are commonly used to collect data in many fields, such as sentiment analysis [35], image recognition [50], and information retrieval [27]. Fig. 1(a) shows a typical microtasks, in which the requester asks the workers to identify the sentiment polarity for a given sentence. In this situation, the requester can easily provide three candidates (positive, negative, and neutral), and the worker chooses the one that he/she believes is right from the candidate options. However, there is a special but common type of task that has not been fully studied. In this task, the requesters demand the workers to list all possible answers for a given question. We denote this type of task as the *listing item task*, which is formally defined in Section 3.1. Fig. 1(b) shows an example of a listing item task in which the requester wants to build a knowledge base about movies and expects to obtain all directors of the movie *Infernal Affairs* by crowdsourcing. In contrast to the former example, the requester cannot provide the candidates for this question. Thus, he/she must release a question such as "Please list all the directors of the movie *Infernal Affairs*" and hopes that the workers could list all the answers that they believe are correct.

Note that the listing item task is relatively common in a variety of scenarios, such as the collection of resource description framework data [1] and the construction of a knowledge base [24]. A statistical analysis by Ipeirotis [22] demonstrated that there were more than 960,000 demands for listing item tasks on Amazon Mechanical Turk from January 2009 to April 2010.

However, previous incentive mechanisms were unsuitable for listing item tasks owing to the following reasons:

1) Most of the previous incentive mechanisms focus only on either the precision [31] or recall [46,47] of the answers, while for listing item tasks, both metrics are equally important. Thus, the incentive mechanism for listing item tasks must take the precision and recall into consideration simultaneously. In the example of Fig. 1(b), the correct answers are "Liu Weiqiang" and "Mai Zhaohui", and the incentive mechanism must encourage the workers to provide both items without any irrelevant content.

2) Recently, the approval mechanism proposed in [39] considers the precision as well as the recall. However, the authors focused only on the choice tasks. In their tasks, there is only one correct answer for each task. In contrast, in listing item tasks, the number of correct answers for each task is uncertain, which makes it impossible to apply the approval mechanism to listing item tasks directly. We made a few adaptable modifications to the approval mechanism (Section 5.5), and (we applied variants to the listing item tasks. However, the experimental results indicated that this approach is not suitable for listing item tasks.

Based on both reasons above, in this paper, we focus on the incentive mechanism for listing item tasks. More specifically, we first provide the formal definition of the listing item task. Then, we propose an effective incentive mechanism for such task. In contrast to the previous incentive mechanisms, the monetary reward for a worker under our incentive mechanism is determined by two factors: 1) the number of submitted correct answers, and 2) the precision of the provided answers. After that, the proposed incentive mechanism is tested to satisfy two criteria: the incentive compatibility and the no free lunch criterion (Section 2.1). The former criterion guarantees the truthfulness of the answers, whereas the latter minimizes the number of spammers (workers who answer questions randomly to earn free rewards). These two criteria are widely used for evaluating the performance of incentive mechanisms [38,39]. Finally, we conduct extensive experiments, and the experimental results demonstrate that our incentive mechanism achieves a remarkable improvement over other mechanisms.

Overall, the main contributions of this paper can be summarized as follows:

- We focus on the incentive mechanism for a special but common type of crowdsourcing task, the listing item task. We provide the definition of the listing item task and propose an incentive mechanism for this task.
- We prove that the proposed mechanism is incentive-compatible and satisfies the no free lunch criterion. The experimental results also demonstrate that our proposed incentive mechanism is effective.

The remainder of this paper is organized as follows. Section 2 introduces the background of i) the incentive mechanism and two criteria that it should satisfy, and ii) a psychological phenomenon called the coarse belief assumption, which is used to design the incentive mechanism. Section 3 presents the definition of the listing item task and the proposed incentive algorithm. Section 4 formalizes the expected reward for a worker and gives mathematical proofs. Section 5 describes the experiment settings, and Section 6 presents the analyses of the experiment results from different perspectives. Section 7 discusses the related work, and finally, Section 8 presents the conclusions drawn and provides directions for future work.

2. Background

2.1. Incentive mechanism

In many cases, the quality of answers provided by different workers can vary considerably. To acquire truthful answers, the requesters must ensure that each worker would provide answers precisely and comprehensively. The rewards for each worker naturally should be different depending on the quality of the answers. Thus, various incentive mechanisms are proposed to motivate workers to complete the questions truthfully in many scenarios [3,30,38]. More specifically, these incentive mechanisms provide strategies for allocating the rewards to the workers involved.

In general, a reasonable incentive mechanism should satisfy the following two criteria [38,39].

Definition 1. Incentive Compatibility. An incentive mechanism is incentive-compatible if it satisfies that the expected reward (Section 4.1) for a worker will be strictly maximized only when he/she answers every question following the expectation of the requesters (Section 3.1).

Using this criterion, the workers are encouraged to submit the answers precisely and comprehensively because only in this way that they obtain maximum monetary rewards. Furthermore, the truthfulness of the answers is largely guaranteed. An analogous criterion is extensively used to prove the rationality of incentive mechanisms [12,39].

Definition 2. No Free Lunch Criterion. The no free lunch criterion means that if the answers provided by a worker are totally wrong, then the reward for that worker would be zero.

In practice, there are many workers who submit a few irrelevant items randomly with the hope of obtaining a few free rewards, and this behavior is prevalent in various crowdsourcing platforms [9,44]. The no free lunch criterion can alleviate the impact of this behavior by giving these workers zero reward.

2.2. Coarse belief assumption

In this paper, we utilize a psychological phenomenon about people's coarse capacity of processing information, which is called the *coarse belief assumption*. This assumption means that the granularity with which people can process information is rough. Miller [32] proposed the assumption that the limit of the average human capacity to process information is seven states. Further studies [16,36,41] complemented various experiments and confirmed this assumption. In addition, Shah et al. [39] also shed some light on this psychological phenomenon in designing the crowdsourcing incentive mechanism.

In this paper, the coarse belief assumption is used to reflect the coarse confidence of a worker for an item. Following the study by Shah et al. [39], we formalize the assumption as follows. We use $p(x)$ to denote the confidence in a worker for an item that he/she provides. Naturally, $p(x)$ is in the range of [0,1]. While according to the coarse belief assumption the granularity at which people can process information is rough, the same goes for the coarse confidence for an item; we can deduce that if $p(x) > 0$, it also exceeds a certain threshold λ. Therefore, the range of $p(x)$ is actually $0 \cup (\lambda, 1]$.

We use the coarse belief assumption to make the confidence be obviously distinguishable. Under this assumption, we can assign a probability value of zero when the confidence of the worker is very low. λ is a hyper-parameter of our incentive mechanism. The goal is to realize an incentive mechanism that is incentive-compatible and satisfies the no free lunch criterion under this assumption.

3. Method description

In this section, we introduce the proposed method. Specifically, we give the definition of the listing item task (Section 3.1) and introduce the details of the incentive algorithm (Section 3.2).

3.1. Listing item task

Formally, we can define a crowdsourcing task as a listing item task if the crowdsourcing task contains the following characteristics:

- There are no candidate options that can be provided by requesters. This characteristic makes workers incapable of completing the listing item task by answering choice questions.
- The number of correct answers of each listing item task is uncertain. Ideally, it can range from 0 to $+\infty$.
- The answers to this task are usually named entities. The named entities are real-world objects, which can be denoted with proper names, such as persons, organizations, and locations. Due to the semantic clarity of the named entities, it is easy for requesters to count the number of each answer provided by the workers. With the statistics of each provided answer, the requesters can simply utilize aggregating algorithms such as majority voting to obtain the final answers for each question.
- High-quality answers for listing item tasks mean that they are precise as well as comprehensive. Because the number of correct answers can exceed one, the expectation of the requesters for listing item tasks is that the workers involved could provide all the items that have a confidence to be correct exceeding 0. Formally, for an arbitrary question k from

N questions ($k \in [1, N]$), a worker responds to the question by listing b items, which we denote as $y_{ki}(i \in [1, b])$. Then, p_{ki} represents the confidence that the worker believes the item y_{ki} to be correct. Because these items are independent, p_{ki} satisfies $0 \le p_{ki} \le 1$. The workers are encouraged to list those items that satisfy the following:

$$\{y_{ki} | p_{ki} > 0\} \tag{1}$$

3.2. Incentive mechanism for the listing item task

We assume that there are $N(N > 0)$ listing item tasks for workers to complete, and $G(1 \le G \le N)$ gold standard questions are injected into the N questions. Gold standard questions are questions for which the requesters have acquired the correct answers in advance. Evidently, the workers do not know which questions are gold standard questions. Because the rewards of the workers are totally dependent on their performance on the gold standard questions, the workers should try hard on every task to obtain the most rewards. Using gold standard questions is an effective strategy for evaluating the quality of answers provided by a worker, and many studies use this method of crowdsourcing to achieve their goals [4,18,49]. We assume that a worker responds with N_w answers for the gold standard questions and that among them, N_c is the number of answers that are proved to be correct. N_g represents the number of correct answers for the gold standard questions.

As introduced before, for listing item tasks, the requester expects workers to list answers with confidence values larger than zero. However, we find that there is no incentive-compatible mechanism that can encourage workers to complete tasks following the expectation of requesters.

Theorem 1. *There is no incentive mechanism that can encourage workers to provide answers precisely and comprehensively, specifically, to list answers with confidence values larger than zero completely following the expectation of requesters.*

The proof of Theorem 1 is presented in the appendix.

Under this situation, we introduce the coarse belief assumption (Section 2.2) to circumvent the result of Theorem 1 and propose our incentive mechanism. The reward of a worker under our incentive mechanism can be calculated by Algorithm 1.

Algorithm 1 Incentive mechanism for listing item tasks.

Input: the number of correct answers N_c, the number of all answers N_w, and the number of gold answers N_g
Output: the worker's reward R

1: Set a maximum reward α by the requester
2: Derive the worker's reward R by

$$f(N_c, N_w) = \alpha \frac{\lambda \frac{N_c}{N_w} + (1 - \lambda) N_c}{\beta} \tag{2}$$

where λ is the balance weight determined by the coarse belief assumption. β is a normalization factor that can be calculated by

$$\beta = \lambda + (1 - \lambda) N_g \tag{3}$$

3: **return** R

In the algorithm, the first step (line 1) is to set the maximum reward α for a worker by the requester. Then, the reward for a worker can be derived by Eq. (2), where λ is the coarse belief threshold introduced in Section 2.2. β is a normalization factor that can be calculated by Eq. (3). According to Eq. (2), the reward is determined by two factors:

1) The precision of the answers provided by the workers N_c/N_w. This factor can be used to avoid "spammers".

2) The number of correct answers N_c. The process involving N_c is important because it can encourage the workers to answer the questions more comprehensively.

Note that the function of β is to make the reward of each worker be in the range of $[0, \alpha]$. The algorithm means that the more correct answers and the higher the precision provided by the worker, the more reward he/she can obtain. Moreover, only when $N_c = N_w = N_g$, which means that the answers are exactly correct, can the reward be α. In addition, no matter how many gold standard questions are injected into the tasks, they can always reflect the quality of answers by this worker. Therefore, the number of gold standard questions does not influence the performance of our algorithm.

Here, we use a simple example to illustrate our method. We assume that there are 100 questions ($N = 100$) released by the requesters. Among them, 10 questions ($G = 10$) are gold standard questions. The workers must answer all the 100 questions, in which the requester utilizes the 10 gold questions to calculate the reward of each worker and hopes to acquire the final answers of the other 90 questions according to the crowdsourcing results, because the workers do not know which ones are the gold standard questions. In this situation, to earn more rewards, they must answer each question following the demands of the requester. Meanwhile, the requester can obtain answers with higher quality for the 90 non-gold standard questions. We assume that there are 25 correct answers ($N_g = 25$) for these 10 gold questions. A worker provides 20 answers ($N_w = 20$), and among them, 15 answers ($N_c = 15$) are correct. If we set $\lambda = 0.3$ and $\alpha = 10$, according to Algorithm 1, the

reward of this worker is as follows:

$$f(15, 20) = 10 \cdot \frac{0.3 \cdot \frac{15}{20} + (1 - 0.3) \cdot 15}{0.3 + (1 - 0.3) \cdot 25} = 6.025$$

In addition, the reward function has the property that when the N_w is fixed, the reward is proportional to N_c. The reward function is formally described in the following theorem:

Theorem 2.

$$f(a, N_w) = af(1, N_w), \ \ 0 \leq a \leq min\{N_g, N_w\}$$

4. Proof for the incentive mechanism

Based on the algorithm introduced above, we can demonstrate that our proposed mechanism is incentive-compatible and satisfies the no free lunch criterion.

Theorem 3. *Our incentive mechanism as presented in Algorithm 1 is incentive-compatible and satisfies the no free lunch criterion.*

In this section, we formalize the expected reward and prove Theorem 3. Moreover, we prove that our incentive mechanism satisfies the two criteria.

4.1. Expected reward

Next, we will formally elaborate the expected reward for the listing item task. We assume that a worker responds to the G gold standard questions by the listing items. The number of items is N_w, and we denote them as $y_1, \ldots, y_{N_w}$. We denote the confidence of a worker with respect to item y_i as the probability to be correct for this item according to this worker (expressed as p_i). Then, we can obtain the corresponding judging flags $\varepsilon_1, \ldots, \varepsilon_{N_w}$, among which $\varepsilon_i \in \{0, 1\}$, $i \in [1, N_w]$, where 1 indicates that the corresponding item y_i is judged to be correct, and 0 to be wrong. Obviously, $N_c = \sum_{i=1}^{N_w} \varepsilon_i$. Then, let E_N denote the expected reward when the worker responds to all of the N listing item tasks, while E_G represents the expected reward when the worker completes the gold standard questions. Because the G gold standard questions are distributed randomly in the N questions, E_N is the summation of each E_G. In addition, $[N]$ represents an integer set $\{1, \ldots, N\}$.

Here, we take a toy example to illustrate the definition of the judging flag ε_i. We assume that a worker provides *"a, b, c, d"* when he/she answers the gold questions. The correct answers for these gold questions are *"a, c, e"*. Then, we can obtain that the judging flags are $\{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4\} = \{1, 0, 1, 0\}$.

Next, we can formalize the expected reward as

$$\begin{aligned} E_N &= \frac{1}{\binom{N}{G}} \sum_{G \subseteq [N]} E_G \\ E_G &= \sum_{N_c, N_w} f(N_c, N_w) p(N_c, N_w) \\ p(N_c, N_w) &= \prod_{i=1}^{N_w} p_i^{1\{\varepsilon_i = 1\}} (1 - p_i)^{1\{\varepsilon_i = 0\}} \end{aligned} \tag{4}$$

In the equations, $f(N_c, N_w)$ is the worker's reward, which is introduced in Section 3.2. The function $p(N_c, N_w)$ represents the probability that when the worker lists N_w items, there are N_c items that are judged to be correct (see Section 4.2).

4.2. Proof for incentive compatibility

The current task is to prove that the proposed incentive mechanism can satisfy incentive compatibility (Definition 1). To explain it much more clearly, the proof is conducted under three situations, from simple to complex, as follows:

(1) The situation that $N = G = 1$.

The simplest situation is $N = G = 1$, which means that there is only one question, and it is a gold standard question. We assume that there are m candidates that a worker believes to be correct, and for the other answers, their correct probabilities are all zero. According to the coarse belief assumption, we can order these probabilities as $p_1 \geq p_2 \geq \cdots \geq p_m > \lambda > p_{m+1} = \cdots = 0$. Theoretically, the worker has infinite strategies when he responds to this question. At the same time, as introduced in Section 3.1, the requester hopes that the worker could provide those m candidates that have a nonzero probability and eliminate the others. According to the definition of incentive compatibility (Definition 1), if the worker takes this action, his/her expected reward will be the highest in all strategies. Therefore, we must first calculate the expected reward when the worker answers those m candidates that have nonzero probabilities. In this situation, the number of correct answers N_c is in the range of $[0, m]$. Next, we will illustrate the reward and the probabilities when N_c changes.

Table 1
Probabilities and rewards of different N_c values when the worker responses the m answers with probability to be correct exceeding 0.

No. of N_c	Probability	Reward
0	$\prod_{i=1}^{m}(1-p_i)$	0
1	$\sum_{i=1}^{m}(p_i\prod_{j=1,j\neq i}^{m}(1-p_j))$	f_{1m}
...	...	...
m	$p_1p_2\cdots p_m$	mf_{1m}

- The case when $N_c = 0$.
  $N_c = 0$ means that none of these m candidates is correct, and the probability of this case is $p(N_c, N_m) = p(0, m) = \prod_{i=1}^{m}(1-p_i)$. According to Eq. (2), the reward is 0.
- The case when $N_c = 1$.
  $N_c = 1$ means that there is one correct answer among the m candidates. Because each of them could be correct, $p(1, m)$ can be calculated by the following summation:

$$\begin{aligned} p(1,m) &= p_1\prod_{j=2}^{m}(1-p_j) + p_2\prod_{j=1,j\neq 2}^{m}(1-p_j) + \cdots + p_m\prod_{j=1}^{m-1}(1-p_j) \\ &= \sum_{i=1}^{m}(p_i\prod_{j=1,j\neq i}^{m}(1-p_j)) \end{aligned}$$

  According to Eq. (2), the reward in this situation is as follows:

$$f(1,m) = \frac{\alpha}{\beta}(\lambda\frac{1}{m} + (1-\lambda))$$

  We denote $f(1, m)$ as f_{1m} for short.
- The case when $N_c = m$.
  Next, we consider the case in which $N_c = m$. When all of the m candidates are verified to be correct, the probability in this case can be expressed as $p(m, m) = p_1p_2\cdots p_m$, and according to Theorem 1, we can present the corresponding reward as $f(m, m) = mf_{1m}$.

The probabilities and rewards for the above cases are included in Table 1, which can be used to calculate the expected reward when the worker submits the m candidate answers. Next, we introduce a theorem.

Theorem 4.

$$p(1,m) + 2p(2,m) + \ldots + mp(m,m) = p_1 + p_2 + \ldots + p_m$$

Proof of Theorem 4.. Here, we prove this theorem using mathematical induction.

First, we consider the condition that when $m = 1$, the corresponding probability is p_1. It is obvious that $p(1, m) = p_1$. Therefore, the equation is true in this condition.

Next, we assume that when $m = k$, the equation is true. Therefore, we obtain the following equation.

$$\begin{aligned} Q_k &= p(1,k) + 2p(2,k) + \ldots + kp(k,k) \\ &= \sum_i(p_i\prod_{j,j\neq i}(1-p_j)) + 2\sum_{i,j,i\neq j}(p_ip_j\prod_{\substack{s,s\neq i,\\ s\neq j}}(1-p_s)) + \ldots + k\prod_i p_i \\ &= p_1 + p_2 + \ldots + p_k \end{aligned}$$

Next, we consider the condition when $m = k + 1$.

$$\begin{aligned} &p(1,k+1) + 2p(2,k+1) + \ldots + (k+1)p(k+1,k+1) \\ &= \sum_i(p_i\prod_{j,j\neq i}(1-p_j)) + 2\sum_{i,j,i\neq j}(p_ip_j\prod_{\substack{s,s\neq i,\\ s\neq j}}(1-p_s)) + \ldots + (k+1)\prod_i p_i \\ &= (1-p_{k+1})(\sum_i(p_i\prod_{j,j\neq i}(1-p_j)) + 2\sum_{i,j,i\neq j}(p_ip_j\prod_{\substack{s,s\neq i,\\ s\neq j}}(1-p_s)) + \ldots + k\prod_i p_i) \\ &+ p_{k+1}(\prod_i(1-p_i) + 2\sum_i(p_i\prod_{j,j\neq i}(1-p_j)) + \ldots + (k+1)\prod_i p_i) \\ &= (1-p_{k+1})Q_k + p_{k+1}\prod_i(1-p_i) + p_{k+1}Q_k + \end{aligned}$$

$$
\begin{aligned}
& p_{k+1}\Big(\sum_i \Big(p_i \prod_{j,j\neq i}(1-p_j)\Big) + \sum_{i,j,i\neq j}\Big(p_i p_j \prod_{\substack{s,s\neq i,\\ s\neq j}}(1-p_s)\Big) + \ldots + \prod_i p_i\Big) \\
&= (1-p_{k+1})Q_k + p_{k+1}\prod_i (1-p_i) + p_{k+1}Q_k + p_{k+1}\Big(1-\prod_i (1-p_i)\Big) \\
&= Q_k + p_{k+1} \\
&= p_1 + p_2 + \ldots + p_k + p_{k+1}
\end{aligned}
$$

Therefore, when $m = k+1$, the equation is correct. Thus, we can obtain that for all m values in the range of $[1, \infty)$, Theorem 2 is true. □

Using this theorem, we can calculate the expected reward when the worker responds to the m candidates as the requester expects:

$$
\begin{aligned}
E_m &= p(1,m)f_{1m} + p(2,m)2f_{1m} + \ldots + p(m,m)mf_{1m} \\
&= f_{1m}(p(1,m) + 2p(2,m) + \ldots + mp(m,m)) \\
&= f_{1m}(p_1 + p_2 + \ldots + p_m)
\end{aligned}
$$

Afterward, we must calculate the expected reward when the worker does not respond to the question as the requester expects. We assume that the worker provides other l answers $\{i_1, i_2, \ldots, i_l\}$. When $l = m$, obviously, the l answers are not exactly the same as the m answers. Thus, they must contain a few items with probabilities of 0. We denote the probabilities of these answers as $p_{i_1}, p_{i_2}, \ldots, p_{i_l}$. The expected reward E_l can be calculated by the following:

$$
\begin{aligned}
E_l &= f_{1l}(p_{i_1} + p_{i_2} + \ldots + p_{i_l}) \\
&= \frac{\alpha}{\beta}(\lambda\frac{1}{l} + (1-\lambda))\left(p_{i_1} + p_{i_2} + \ldots + p_{i_l}\right) \\
&= \frac{\alpha}{\beta}(\lambda\frac{1}{m} + (1-\lambda))\left(p_{i_1} + p_{i_2} + \ldots + p_{i_l}\right) \\
&< f_{1m}(p_1 + p_2 + \ldots + p_m) \\
&= E_m
\end{aligned}
$$

It can be observed that when the worker provides the other l answers $(l = m)$, the expected total reward E_l is strictly smaller than E_m.

Next, we consider the case when $l > m$, because there are only m answers with probabilities greater than λ. Therefore, the l answers in this case must contain a few answers with a probability of 0. Then, the expected reward can be calculated as follows:

$$
\begin{aligned}
E_l &= f_{1l}\left(p_{i_1} + p_{i_2} + \ldots + p_{i_l}\right) \\
&\leq f_{1l}(p_1 + p_2 + \ldots + p_m) \\
&= \frac{\alpha}{\beta}(\lambda\frac{1}{l} + (1-\lambda))(p_1 + p_2 + \ldots + p_m) \\
&< \frac{\alpha}{\beta}(\lambda\frac{1}{m} + (1-\lambda))(p_1 + p_2 + \ldots + p_m) \\
&= f_{1m}(p_1 + p_2 + \ldots + p_m) \\
&= E_m
\end{aligned}
$$

It is obvious that E_l in this case is also strictly smaller than E_m.

Finally, we consider the case in which the worker lists l answers $(l < m)$. Obviously, the probabilities of a few answers in these l answers could exceed 0, and we denote the sum of these probabilities as p_A, which satisfies $p_A \leq l$. The probabilities of the remaining l answers are all 0. In addition, there could be a few answers with probabilities exceeding 0, but they are not listed in these l answers. Thus, we present the sum of these probabilities as p_B, which satisfies $p_B > \lambda(m-l)$. Then, the expected reward is as follows:

$$
\begin{aligned}
E_l &= f_{1l}p_A \\
&= \frac{\alpha}{\beta}(\lambda\frac{1}{l} + (1-\lambda))p_A \\
&= \frac{\alpha}{\beta}((\lambda\frac{1}{m} + (1-\lambda)(p_A + p_B)) + \lambda(\frac{1}{l} - \frac{1}{m})p_A - (\lambda\frac{1}{m} + (1-\lambda))p_B) \\
&= E_m + \frac{\alpha}{\beta}(\lambda(\frac{1}{l} - \frac{1}{m})p_A - (\lambda\frac{1}{m} + (1-\lambda))p_B)
\end{aligned}
$$

$$< E_m + \frac{\alpha}{\beta}(\lambda(\frac{1}{l} - \frac{1}{m})l - (\lambda\frac{1}{m} + (1-\lambda))\lambda(m-l))$$
$$= E_m + \frac{\alpha}{\beta}(\frac{l}{m}(\lambda^2 - \lambda) + (m-l-1)(\lambda^2 - \lambda))$$
$$< E_m$$

Similarly, the expected reward in this case is strictly smaller than E_m.

Above all, we have proved that when $N = G = 1$, the incentive mechanism is incentive-compatible.

(2) The situation in which $N = G \geq 1$.

Next, we consider the situation in which $N = G \geq 1$. According to Eq. (4), only when a worker responds to every question as the requester expects can he/she obtain the highest expected reward. Therefore, the expected reward in this case also satisfies incentive compatibility.

(3) The situation in which $N > G \geq 1$.

Finally, we consider the situation where $N > G \geq 1$. By considering the general expected reward from Eq. (4), we proved above that E_G is incentive-compatible, and it is obvious that the general total expected reward E_N will be maximized if each E_G is maximized. In other words, for a series of listing item tasks, if the worker completes every question as the requester expects, his/her reward will be maximized.

Thus, the incentive mechanism proposed in this paper is proved to be incentive-compatible.

4.3. Proof for no free lunch criterion

As introduced in Section 2.1, a qualified incentive mechanism must also satisfy the no free lunch criterion. Depending on Algorithm 1, if $N_c = 0$, which means that a worker responds wrongly to all the gold standard questions, then his/her final payment will be zero. Obviously, our incentive mechanism satisfies the no free lunch criterion.[1]

5. Experimental settings

In this section, we present the settings of the experiments.

5.1. Design of tasks

We designed 50 listing item tasks similar to the example of Fig. 1(b). These tasks are selected from various domains. There are 18 tasks from geography, 13 tasks from culture, 10 tasks from history, 5 tasks from engineering, and 4 tasks from biology. In addition, the number of gold answers ranges from 2 to 13, and the average number of gold answers is 4.8.

In general, the ratio of the gold standard questions to all of the questions should be 1/100 at a minimum [34]. However, in the experiments, to evaluate the quality of answers, the correct answers for all tasks have been known in advance.

5.2. Platforms and workers

We conducted the experiments on two different crowdsourcing platforms, CrowdKnow and ZhiDao.[2] CrowdKnow is a crowdsourcing platform in our school, whereas ZhiDao is the biggest Chinese crowdsourcing platform.[3]

1) For CrowdKnow, we recruited 60 college students and assigned them to 6 different incentive mechanisms randomly (i.e., fixed payment mechanism, precision mechanism, recall mechanism, two variants of the approval mechanism in [39], and our mechanism; see Section 5.5; on average, there are 10 students in each mechanism). All of the students are familiar with the domains of the tasks and can understand the different mechanisms. The demographics of these students are depicted in Table 2.

2) For ZhiDao, we posted the same questions and illustrated our demands. After collecting the answers from 10 workers under each mechanism, we close the tasks.

5.3. Evaluation metrics

As introduced in Section 3.1, for the listing item task, the requirement of the requesters is to obtain precise and comprehensive answers. Thus, we used the $F1$ score to measure the quality of answers. The $F1$ score is widely used in various fields, such as machine learning [23] and information retrieval [2,25].

The $F1$ score is the harmonic mean of precision and recall [6,37]:

$$F1\ score = 2 \cdot precision \cdot recall / (precision + recall) \tag{5}$$

[1] In this section, we have proved that our incentive mechanism is incentive-compatible and satisfies the no free lunch criterion. However, it is better to prove the optimality of our method in all possible incentive mechanisms, but we leave this for future work.

[2] zhidao.baidu.com

[3] All tasks and collected answers from these platforms can be found at https://github.com/sophie996/Incentive-mechanism.

Table 2
Demographics of students on CrowdKnow.

Category	Participants	Percentage
Total	60	100
Undergraduates	36	60.0
Postgraduates	19	31.7
Others	5	8.3
Science and engineering	22	36.7
Liberal arts	30	50.0
Others	8	13.3
Familiar with crowdsourcing	58	96.7
Does not know crowdsourcing	2	3.3
Familiar with the domains of the tasks	60	100
Not familiar with the domains of the tasks	0	0
Understand the incentive mechanism clearly	60	100
Cannot understand the incentive mechanism	0	0

The larger the $F1$ score, the higher the quality of answers.

5.4. Evaluating objects

We evaluated the quality of answers from two perspectives:

1) **The perspective of the original answers.** We evaluate the $F1$ score of the original answers provided by each worker. We assume that the answers provided by a worker are *"a, b, c, d"* and the gold answers are *"a, c, e"*. Then, the precision is 0.5, and the recall is 0.67. According to Eq. (5), $F1\ score = 2 \cdot 0.5 \cdot 0.67/(0.5 + 0.67) = 0.57$.

2) **The perspective of the final answers.** To obtain the final answers for a crowdsourcing task, we always assign the same task to multiple workers and then aggregate their respective answers. Majority voting is a common way of aggregating the answers [21,43]. We assume that there are 5 workers to complete a task, and all of the answers are *"a, a, a, a, b, c, c, c, d, e, f"*. In majority voting, only when the answers are provided for more than μ times that they can be chosen as the final answers for this task, where μ is the hyper-parameter. In the example, if $\mu = 3$, the final answers for this task are *"a, c"*. If the gold answers are *"a, c, e"*, then the precision is 1.0, the recall is 0.67, and the $F1\ score = 2 \cdot 1.0 \cdot 0.67/(1.0 + 0.67) = 0.80$.

5.5. Comparing incentive mechanisms

Here, we compared the performance of the following six mechanisms.

1) **Fixed payment mechanism.** Workers will obtain a fixed payment when they complete the tasks regardless of the quality of their answers. Many studies use this mechanism as a baseline to evaluate the results [10,18].

2) **Precision mechanism.** In this mechanism, the reward of a worker is totally determined by the precision, i.e., $f_p(N_c, N_w) = \alpha N_c/N_w$ [31].

3) **Recall mechanism.** In the recall mechanism, the reward is totally determined by the recall, i.e., $f_r(N_c) = \alpha N_c/N_g$. References [19,46,47] all chose this mechanism as the incentive mechanism.

4) **Approval mechanism.** This mechanism is the state-of-the-art mechanism for choice tasks [39]. The approval mechanism considers the precision and recall simultaneously. In their work, the workers are encouraged to select all of the possible correct options for each task. Note that the main characteristic of the approval mechanism is that the number of gold answers for each task is only one, and only when the options selected by a worker for each task all contain the gold answer can this worker obtain the reward. Otherwise, the reward would be zero, even if the worker incorrectly answers only one task while correctly answering the others. Here, we are curious about the performance when we apply the approval mechanism to listing item tasks. Because the approval mechanism is designed for choice tasks with only one gold answer, it only considers whether the selected options contain the gold answer or not; thus, it cannot be applied to listing item tasks directly. Therefore, two relevant variant mechanisms are introduced as follows. a) **Variant 1: Loose approval mechanism.** If a worker provides answers for every task that contain at least one gold answer, he/she will receive the reward. We define this criterion as the loose criterion. Specifically,

$$f_{la}(N_c, N_w) = \begin{cases} \alpha(1-\lambda)^{(N_w - N_c)}, & answers\ satisfy\ the\ loose\ criterion \\ 0, & otherwise \end{cases} \tag{6}$$

b) **Variant 2: Strict approval mechanism.** If a worker provides answers for every task that contain all of the gold answers, he/she will obtain the reward. We denote this criterion as the strict criterion. Specifically,

$$f_{sa}(N_c, N_w) = \begin{cases} \alpha(1-\lambda)^{(N_w - N_g)}, & answers\ satisfy\ the\ strict\ criterion \\ 0, & otherwise \end{cases} \tag{7}$$

In the variants above, we set the coarse belief threshold as $\lambda = 0.3$.

Table 3
Quality of original answers from workers.

Platform	No.	Mechanism	*F*1 score	Precision	Recall
CrowdKnow	1	Fixed payment mechanism	0.425	0.766	0.327
	2	Precision mechanism	0.386	0.855	0.270
	3	Recall mechanism	0.675	0.642	**0.762**
	4	Loose approval mechanism	0.362	**0.872**	0.242
	5	Strict approval mechanism	0.592	0.521	0.748
	6	Our mechanism	**0.727**	0.804	0.694
ZhiDao	7	Fixed payment mechanism	0.393	0.771	0.285
	8	Precision mechanism	0.552	**0.871**	0.433
	9	Recall mechanism	0.603	0.574	0.67
	10	Loose approval mechanism	0.473	0.842	0.352
	11	Strict approval mechanism	0.606	0.545	**0.746**
	12	Our mechanism	**0.724**	0.844	0.669

5) **Our mechanism.** This mechanism is our proposed mechanism indicated in Algorithm 1, where the reward of a worker is determined by the precision and recall. Following the former mechanisms, the coarse belief threshold λ was set as 0.3.

6. Experimental results

In this section, we present the experimental results and analyze them in detail.[4]

6.1. Analysis of original answers

Table 3 presents the results of the original answers from workers under the different mechanisms. We can analyze the results from the following aspects.

Our mechanism vs. fixed payment mechanism. As indicated in row 1 and row 6 in Table 3, it is evident that our mechanism outperforms the fixed payment mechanism. Specifically, the *F*1 score improves by 0.302 (0.425 vs. 0.727), and the precision and recall of our mechanism are also higher than those of the fixed payment mechanism. Thus, our mechanism can better encourage workers to provide high-quality answers than the fixed payment mechanism.

Our mechanism vs. precision mechanism and recall mechanism. Because many mechanisms that incentivize workers depend on only the precision or recall of the answers, we also conducted experiments to compare the performance of our mechanism with the precision mechanism and recall mechanism. The results are presented in rows 2, 3, and 6 in Table 3. We can conclude that our mechanism can more incentivize workers to provide answers with higher quality than the other two mechanisms (the *F*1 scores under the precision mechanism and recall mechanism are 0.386 and 0.675, respectively, whereas the *F*1 score under our mechanism is 0.727). Naturally, the precision mechanism tends to encourage workers to submit high-precision answers that are, at the same time, low-recall answers. In contrast, the recall mechanism produces high-recall answers that are, at the same time, low-precision answers. These results demonstrate that precision and recall are both important. Our mechanism, which considers both precision and recall, can effectively incentivize workers to submit more balanced answers.

Our mechanism vs. approval mechanism. We also compared the quality of answers under our mechanism and the approval mechanism. The results demonstrate the superiority of our mechanism (in rows 4 and 5, the *F*1 scores are 0.362 and 0.592 under the two variants of the approval mechanism, and 0.727 under our mechanism). Furthermore, we can draw the following conclusions: i) The performance of the loose approval mechanism is similar to that of the precision mechanism. The precision scores are both high (0.872 and 0.855, respectively), while the recall scores are all low (0.242 and 0.270, respectively). The reason could be that under the loose approval mechanism, workers can obtain the reward once the answers contain more than one gold answer for each task. Therefore, the workers tend to submit answers with the highest probability to be correct. ii) The strict approval mechanism is similar to the recall mechanism. The recall scores under these two mechanisms are very high (0.748 and 0.762), while their precision scores are both slightly lower than those of the other mechanisms (0.521 and 0.642). We think that this could be because the workers are required to provide all of the gold answers for each task under the strict approval mechanism. Consequently, the workers must list as many answers as possible to ensure that all of the gold answers are included.

From the results, we can conclude that the approval mechanism is not suitable for listing item tasks. Our mechanism can obtain more comprehensive and precise answers.

[4] The participation duration of the experiments is 3 days on CrowdKnow and 8 days on ZhiDao.

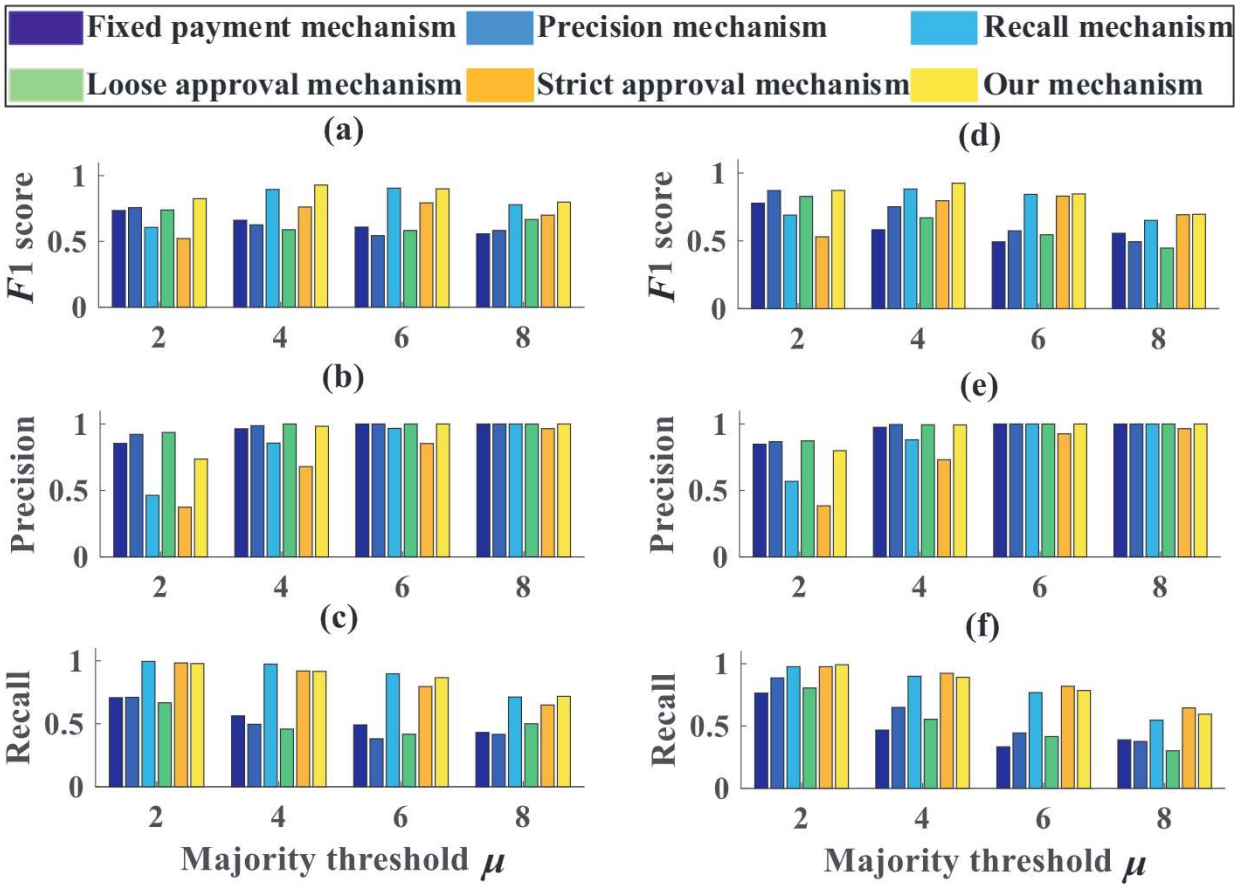

Fig. 2. Results of final answers. The left part shows the (a) *F*1 score, (b) precision, and (c) recall on CrowdKnow. The right part indicates the (d) *F*1 score, (e) precision, and (f) recall on ZhiDao. On all of the subfigures, the *X*-axis denotes the different majority thresholds μ, whereas the *Y*-axis indicates the *F*1 score, precision, and recall. As shown, the precision mechanism and loose approval mechanism can obtain more precise results. The recall mechanism and strict approval mechanism can obtain more comprehensive results. The *F*1 score of our mechanism has the highest values in most cases, indicating that our method is efficient.

6.2. Analysis of final answers

In this experiment, we aggregated the original answers for the tasks with different majority thresholds and analyzed the results. Fig. 2 illustrates the results of the final answers under the different mechanisms.

From Fig. 2, we can draw the following conclusions:

1) The quality of the final answers is similar to the original answers under each mechanism. Specifically, we consider the strict approval mechanism and recall mechanism: when the majority threshold μ is small, their recall scores are high, while the precision scores are low. With the increase in μ, the precision scores of the two mechanisms increase. We think that this is because the majority of thresholds can help to filter a few incorrect answers.
2) The loose approval mechanism is similar to the precision mechanism in terms of the quality of the final answers. The final answers have high precision scores and low recall scores.
3) From the results, the *F*1 score of our mechanism has the highest values in most cases, which proves that our mechanism can obtain answers with high precision and recall simultaneously.

In summary, our mechanism performs better than the fixed payment mechanism on both the precision and recall. The precision mechanism and recall mechanism can obtain answers with high precision or high recall, which cannot ensure precise and comprehensive answers. The approval mechanism is not suitable for listing item tasks. Our mechanism is effective as it can obtain answers with high *F*1 scores.

6.3. Mean rewards of different mechanisms

We examined the differences in the rewards of the various mechanisms by analyzing the mean values. Table 4 presents the mean rewards under the different mechanisms. The results demonstrate that the reward of our mechanism is slightly

Table 4
Mean rewards of different mechanisms.

Platform	Mechanism	Mean reward
CrowdKnow	Fixed payment mechanism	10.000
	Precision mechanism	8.482
	Recall mechanism	6.840
	Our mechanism	**6.132**
ZhiDao	Fixed payment mechanism	10.000
	Precision mechanism	8.459
	Recall mechanism	6.154
	Our mechanism	**6.100**

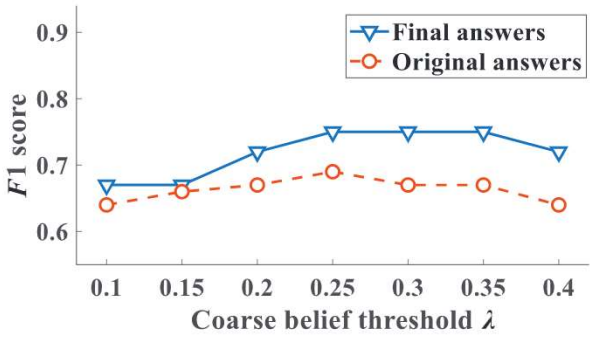

Fig. 3. *F*1 scores of answers with different coarse belief thresholds λ.

lower than that of the others, which proves that our mechanism can improve the quality of answers not because of the amount of reward.

Therefore, our mechanism can obtain answers with high quality at the least amount of reward.

6.4. Effects of coarse belief threshold λ

Because the coarse belief threshold λ is a hyper-parameter of our mechanism, we also conducted an experiment to evaluate the effect of λ on the quality of answers. Additional 35 college students were recruited to conduct the experiment.[5]

Fig. 3 depicts the results, in which the *X*-axis presents different coarse belief thresholds and the *Y*-axis indicates the average *F*1 score of the answers (the solid line is for the final answers, and the dashed line is for the original answers; with the majority threshold $\mu = 3$). As shown in the figure, when λ increases from 0.1 to 0.4, the *F*1 scores of the original answers and final answers have a slight change. For example, the gap between the maximum and minimum *F*1 scores of the original answers is only 0.05 (0.69 vs. 0.64). The results demonstrate that the coarse belief threshold λ has little influence on the performance of our method. The reason could be that the workers usually have only a rough understanding of the mechanism. They learn that the reward is related to both the precision and recall; however, the value of the parameter has little effect on the behavior of the workers.

6.5. Evaluation of the validity of the incentive mechanism

A few researchers found that incentive mechanisms are effective only in specific cases [18–20,40,47,48]. Ho et al. [19] proved that only when tasks are effort-responsive can the quality-based incentive mechanism be effective. An effort-responsive task means that if a worker exerts more effort on the task, the quality of the answers will be higher. The time spent by a worker on a task is a proxy measure of the effort.

Because the experimental results above demonstrate that our incentive mechanism can improve the quality of answers, we are curious about whether the finding by Ho et al. [19] is applicable to listing item tasks. Thus, we conducted an experiment according to the theory in [19]. Fig. 4 illustrates the correlation of the time the workers spent and the *F*1 score of the answers. The *X*-axis is the time when the 5 workers completed the listing item task, and the *Y*-axis denotes the *F*1 score of the answers. The regression of the *F*1 scores demonstrates that with time, the *F*1 score of the quality improves. The results indicate that the listing item task is effort-responsive. Consequently, according to the finding in [19], the quality of the answers for listing item tasks can be improved through incentive mechanisms.

[5] In this experiment, seven different λ values were chosen, as shown in Fig 3. Five different workers were asked to complete the tasks under each λ.

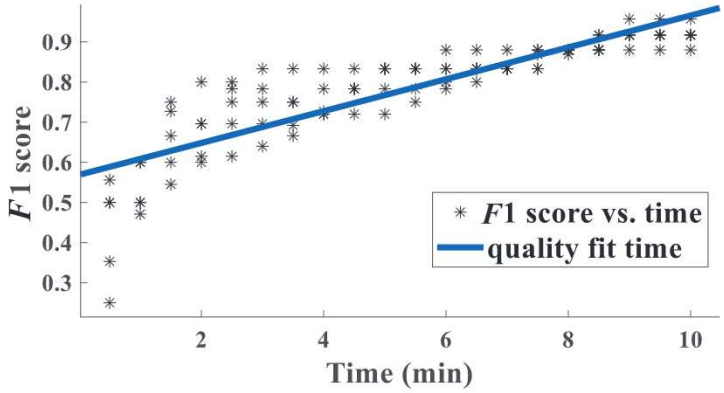

Fig. 4. *F*1 score of answers vs. time for the listing item task. The blue line is the regression line with 95% confidence level. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

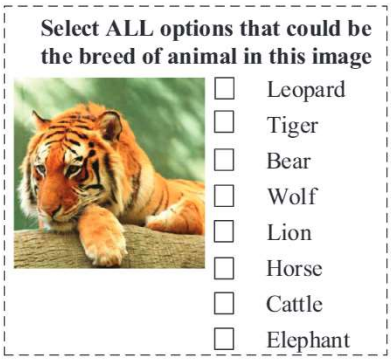

Fig. 5. Example of a multiple-choice task.

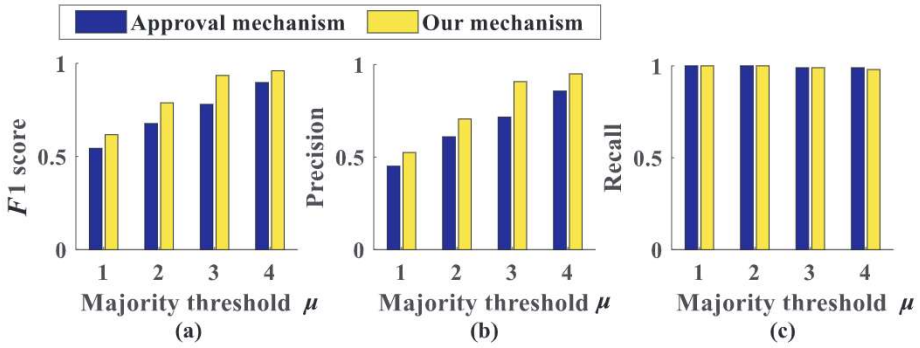

Fig. 6. Results of final answers for multiple-tasks. As shown, the recall values by the approval mechanism and our mechanism are close to 1, but the precision by our mechanism is higher than that of the approval mechanism. Therefore, our mechanism can obtain more precise answers.

6.6. Performance on multiple-choice tasks

As introduced before, the approval mechanism in [39], which is proposed for multiple-choice tasks, also considers the accuracy of answers. Therefore, we are also interested in applying our mechanism to multiple-choice tasks.

We applied our incentive mechanism to the multiple-choice tasks in [39] and compared the performance with the approval mechanism in [39]. Different from the listing item task, requesters provide several candidate options for each multiple-choice task, and workers complete each task by simply selecting a few options. The number of correct answers for each multiple-choice task is only one.

We designed 100 multiple-choice tasks, and an example is shown in Fig. 5. The tasks were issued on our crowdsourcing platform. There were 20 workers completing these tasks, and 10 of them were incentivized by the mechanism in [39] while the others were incentivized by our mechanism. The experimental results are illustrated in Fig. 6.

Table 5
Quality of original answers from workers for multiple-choice tasks.

Mechanism	*F*1 score	Precision	Recall
Approval mechanism	0.653	0.568	0.962
Our mechanism	0.753	0.663	0.979

The *X*-axis of the figures denotes different majority thresholds, whereas the *Y*-axis depicts the average (a) *F*1 score, (b) precision, and (c) recall of aggregated answers.

All of the recall values by the two methods are above 0.98, which means that the correct answers can be selected by workers under the two methods. However, the precision values of the two methods vary significantly. The maximum difference between the precision of the two methods is 0.192 (when the majority threshold is equal to 6, the precision by the approval mechanism is 0.716, whereas the precision by our mechanism is 0.908). The results mean that our mechanism can incentivize workers to select the correct answers and reduce the number of wrong answers simultaneously. This could be because the approval mechanism is strict and makes workers more conservative: once a correct answer for any task is not selected, the total reward of this worker will be zero. Thus, to ensure that the correct answer is selected for every task, workers prefer to select more incorrect answers, which may lead to too much noise in the collected answers. In contrast, our mechanism does not set the final reward as zero when workers complete one task incorrectly.

The average *F*1 score, precision, and recall of the workers are presented in Table 5. We can draw the same conclusion: both methods can incentivize workers to select answers with high recall; however, our mechanism can obtain answers with less noise than the approval mechanism.

From the above results, we can conclude that our mechanism can also be applied to multiple-choice tasks, and it performs better than the approval mechanism.

7. Related work

In this paper, we proposed the incentive mechanism for a special type of task, i.e., the listing item task. To the best of our knowledge, this paper is the first one to address the incentive mechanism for listing item tasks. In fact, there have been several outstanding studies that addressed the incentive mechanisms for crowdsourcing tasks. Among them, the most related one is the study by Shah et al. [39]. In their method, they focused on the incentive mechanism for multiple-choice tasks and evaluated the quality while considering both the precision and recall. However, the experiments proved that their method is not suitable for listing item tasks.

A few studies evaluated the quality depending only on either the precision or the recall. Ho et al. [19] analyzed a series of incentive mechanisms on article proofreading tasks and proved the effectiveness of the mechanisms on such tasks. Similarly, the authors in [47] and [46] demonstrated the effects of different incentive mechanisms on the quality of spotting differences tasks and puzzle game tasks. However, these three studies utilized only the recall as the metric. Mao et al. [31] designed an incentive mechanism for citizen science tasks, and the precision was considered to be the unique metric. Similarly, Shah et al. [38] designed an incentive mechanism for image annotation. These two methods took only the precision into consideration. However, both the precision and recall are important for listing item tasks. In addition, our experiments demonstrate that our proposed method outperforms the precision mechanism and recall mechanism.

Chen et al. [13] proposed an incentive method for information sharing with the help of an additional system called reputation system. However, in most of the current crowdsourcing platforms, the reputation system is not always available. A few systems ask the workers to provide extra information, such as predicting the answers of other workers [15,33]. Xie et al. [45] proposed an incentive strategy by requiring the requesters to grade every answer. Blohm et al. [7] focused on an evaluation using the rating scale and the preference market. These methods are labor-intensive and tend to be ineffective when the number of tasks is very large. Zhai et al. [49] proposed an incentive strategy on crowdsourcing in the clinical natural language processing field with the goal of reducing the monetary cost. In our method, we expect to improve the quality of answers within a fixed budget.

A few studies focused on the validity of incentive mechanisms in different cases [18–20,40,48]. In [19], the authors verified that the performance-based incentive mechanisms are effective only for effort-responsive tasks. In this paper, we conducted experiments and proved that the listing item task is effort-responsive, and furthermore, we confirm the validity of our incentive mechanism.

Actually, apart from the incentive mechanisms, there are still other methods that can improve the quality of answers. Leimeister et al. [28] enhanced the crowdsourcing outcomes by implementing ideas competition. Aroyo and Welty [4] ensured the quality of crowdsourcing annotation by selecting qualified workers and using effective approaches for aggregating the results. Although these methods did not focus on incentive mechanisms, they have also inspired our method.

Overall, the current methods are not suitable for listing item tasks. Therefore, we concentrated on this type of crowdsourcing task and proposed an automatic incentive mechanism to improve the quality of answers.

8. Conclusions and future work

In this paper, we focused on a common type of crowdsourcing task, which is formally defined as the listing item task. Then, we proposed an incentive mechanism and proved that the mechanism is incentive-compatible and satisfies the no free lunch criterion theoretically. Finally, we conducted extensive experiments, and the results demonstrated the superior performance of our incentive mechanism on listing item tasks.

Although we have proved that our incentive mechanism satisfies the two criteria, i.e., incentive compatibility and no free lunch, we have not yet proven the optimality of our method in all possible incentive-compatible mechanisms. Therefore, this will be the topic for our next research. In addition, besides the listing item tasks, there are still many other types of subjective tasks, such as sentence tasks and paragraph tasks, which are commonly used in various fields, i.e., man-machine dialogue, translation, and summary. In the future, we will further extend the incentive mechanism for these tasks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under grant no. 61672102, no. 61073034, no. 61370064 and no. 60940032; the National Social Science Foundation of China under grant no. BCA150050; the Program for New Century Excellent Talents in the University of Ministry of Education of China under grant no. NCET-10-0239; and the Open Project Sponsor of Beijing Key Laboratory of Intelligent Communication Software and Multimedia under grant no. ITSM201493.

Appendix

Proof of Theorem 1

Here, we assume that there exists an incentive mechanism that is incentive-compatible and can encourage workers to provide answers following the expectation of requesters. We denote the payment function as $f(N_c, N_w)$, where N_c is the number of correct answers provided by a worker, and N_w is the number of all answers provided by this worker. We prove the theorem through contradiction discussion.

First, we consider a simple case in which $N = G = 1$. Because the expectation of requesters is that workers can provide answers precisely and comprehensively, for an incentive-compatible mechanism, the reward need to satisfy:

$$f(1,1) > f(1,2)$$
$$f(2,2) > f(1,2)$$

We assume that a worker considers two answers a_1 and a_2, and the confidence values are p_1 and p_2, respectively. When $p_2 = 1$ and $p_1 \in (0, 1]$, the worker is required to provide a_1 and a_2 instead of a_2 alone. Therefore, we need

$$p_1 f(2,2) + (1 - p_1) f(1,2) > f(1,1)$$

Then, we can deduce that

$$p_1 > \frac{f(1,1) - f(1,2)}{f(2,2) - f(1,2)} \tag{8}$$

The result means that only when p_1 satisfies Eq. (8) can the worker be encouraged as the requester expects, and when $p_1 \in (0, \frac{f(1,1)-f(1,2)}{f(2,2)-f(1,2)})$, the incentive mechanism cannot ensure the worker to provide answers as expected. Here, we get a contradiction.

Next, we extend to a general case. When $N \geq G \geq 1$, the worker has provided N_w answers, and among which, N_c answers are correct. For an incentive-compatible mechanism, the reward needs to satisfy

$$f(N_c, N_w) > f(N_c, N_w + 1)$$
$$f(N_c + 1, N_w + 1) > f(N_c, N_w + 1)$$

When the worker considers a more answer with confidence denoted as p, $p \in (0, 1]$. We need

$$p f(N_c + 1, N_w + 1) + (1 - p) f(N_c, N_w + 1) > f(N_c, N_w)$$

Then, we can obtain

$$p > \frac{f(N_c, N_w) - f(N_c, N_w + 1)}{f(N_c + 1, N_w + 1) - f(N_c, N_w + 1)} \tag{9}$$

From Eq. (9), we can find that when $p \in (0, \frac{f(N_c,N_w)-f(N_c,N_w+1)}{f(N_c+1,N_w+1)-f(N_c,N_w+1)})$, the incentive mechanism cannot encourage the worker to provide answers as desired. Here, we also obtain a contradiction. Thus, Theorem 1 is proved.

References

[1] M. Acosta, E. Simperl, F. Flck, M.E. Vidal, R. Studer, Rdf-hunter: automatically crowdsourcing the execution of queries against rdf data sets, Comput. Sci. 7695 (2015) 212–226.
[2] O. Alhabashneh, R. Iqbal, F. Doctor, A. James, Fuzzy rule based profiling approach for enterprise information seeking and retrieval, Inf. Sci. 394 (2017) 18–37.
[3] J. Almenberg, K. Kittlitz, T. Pfeiffer, An experiment on prediction markets in science, PloS One 4 (12) (2009) e8500.
[4] L. Aroyo, C. Welty, Crowd truth: harnessing disagreement in crowdsourcing a relation extraction gold standard, IBM Res. (2013).
[5] Y. Baba, H. Kashima, Statistical quality estimation for general crowdsourcing tasks, in: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013, pp. 554–562.
[6] R. Baeza-Yates, B. Ribeiro-Neto, et al., Modern Information Retrieval, 463, ACM press New York, 1999.
[7] I. Blohm, C. Riedl, J. Füller, J.M. Leimeister, Rate or trade? identifying winning ideas in open idea sourcing, Inf. Syst. Res. 27 (1) (2016) 27–48.
[8] I. Blohm, S. Zogaj, U. Bretschneider, J.M. Leimeister, How to manage crowdsourcing platforms effectively? California Manage. Rev. 60 (2) (2018) 122–149.
[9] J. Bohannon, Social science for pennies, Science 334 (6054) (2011). 307–307
[10] K.J. Boudreau, N. Lacetera, K.R. Lakhani, Incentives and problem uncertainty in innovation contests: an empirical analysis, Manage. Sci. 57 (5) (2011) 843–863.
[11] D.C. Brabham, Crowdsourcing as a model for problem solving: an introduction and cases, Convergence 14 (1) (2008) 75–90.
[12] R. Cavallo, S. Jain, Efficient crowdsourcing contests, in: Proceedings of International Conference on Autonomous Agents and Multiagent Systems, 2012, pp. 677–686.
[13] J. Chen, H. Xu, A.B. Whinston, Moderated online communities and quality of user-generated content, J. Manage. Inf. Syst. 28 (2) (2011) 237–268.
[14] A.I. Chittilappilly, L. Chen, S. Amer-Yahia, A survey of general-purpose crowdsourcing techniques, IEEE Trans. Knowl. Data Eng. 28 (9) (2016) 2246–2266.
[15] A. Dasgupta, A. Ghosh, Crowdsourced judgement elicitation with endogenous proficiency, in: Proceedings of International Conference on World Wide Web, 2013, pp. 319–330.
[16] J.L. Doumont, Magical numbers: the seven-plus-or-minus-two myth, IEEE Trans. Prof. Commun. 45 (2) (2002) 123–127.
[17] D. Haas, J. Ansel, L. Gu, A. Marcus, Argonaut: macrotask crowdsourcing for complex data processing, Proc. VLDB Endow. 8 (12) (2015) 1642–1653.
[18] C. Harris, You're hired! an examination of crowdsourcing incentive models in human resource tasks, in: Proceedings of ACM WSDM Workshop on Crowdsourcing for Search and Data Mining, 2011, pp. 15–18.
[19] C.-J. Ho, A. Slivkins, S. Suri, J.W. Vaughan, Incentivizing high quality crowdwork, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 419–429.
[20] C.-J. Ho, A. Slivkins, J.W. Vaughan, Adaptive contract design for crowdsourcing markets: bandit algorithms for repeated principal-agent problems, J. Artif. Intell. Res. 55 (2016) 317–359.
[21] N.Q.V. Hung, N.T. Tam, L.N. Tran, K. Aberer, An evaluation of aggregation techniques in crowdsourcing, in: Proceedings of International Conference on Web Information Systems Engineering, 2013, pp. 1–15.
[22] P.G. Ipeirotis, Analyzing the amazon mechanical turk marketplace, XRDS 17 (2) (2010) 16–21.
[23] F.H. Khan, U. Qamar, S. Bashir, Esap: a decision support framework for enhanced sentiment analysis and polarity classification, Inf. Sci. 367 (2016) 862–873.
[24] Y. Kim, W. Jung, K. Shim, Integration of graphs from different data sources using crowdsourcing, Inf. Sci. 385 (2017) 438–456.
[25] O. Kolomiyets, M.-F. Moens, A survey on question answering technology from an information retrieval perspective, Inf. Sci. 181 (24) (2011) 5412–5434.
[26] P. Kucherbaev, F. Daniel, S. Tranquillini, M. Marchese, Crowdsourcing processes: a survey of approaches and opportunities, IEEE Internet Comput. 20 (2) (2016) 50–56.
[27] J. Lee, D. Lee, S.-w. Hwang, Crowdk: answering top-k queries with crowdsourcing, Inf. Sci. 399 (2017) 98–120.
[28] J.M. Leimeister, M. Huber, U. Bretschneider, H. Krcmar, Leveraging crowdsourcing: activation-supporting components for it-based ideas competition, J. manage. inf. Syst. 26 (1) (2009) 197–224.
[29] G. Li, J. Wang, Y. Zheng, M.J. Franklin, Crowdsourced data management: a survey, IEEE Trans. Knowl. Data Eng. 28 (9) (2016) 2296–2319.
[30] S. Luckner, Prediction markets: how do incentive schemes affect prediction accuracy? in: Proceedings of Dagstuhl Seminar, 2007.
[31] A. Mao, E. Kamar, Y. Chen, E. Horvitz, M.E. Schwamb, C.J. Lintott, A.M. Smith, Volunteering versus work for pay: incentives and tradeoffs in crowdsourcing, in: First AAAI Conference on Human Computation and Crowdsourcing, 2013.
[32] G.A. Miller, The magical number seven, plus or minus two: some limits on our capacity for processing information., Psychol. Rev. 63 (2) (1956) 81.
[33] N. Miller, P. Resnick, R. Zeckhauser, Eliciting informative feedback: the peer-prediction method, Manage. Sci. 51 (9) (2005) 1359–1373.
[34] D. Oleson, A. Sorokin, G. Laughlin, V. Hester, J. Le, L. Biewald, Programmatic gold: targeted and scalable quality assurance in crowdsourcing, in: Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.
[35] B. Pang, L. Lee, et al., Opinion mining and sentiment analysis, Found. Trends® Inf. Retr. 2 (1–2) (2008) 1–135.
[36] T.L. Saaty, M.S. Ozdemir, Why the magic number seven plus or minus two, Math. Comput. Model. 38 (3) (2003) 233–244.
[37] Y. Sasaki, et al., The truth of the f-measure, Teach Tutor Mater 1 (5) (2007) 1–5.
[38] N.B. Shah, D. Zhou, Double or nothing: multiplicative incentive mechanisms for crowdsourcing, in: Proceedings of Advances in Neural Information Processing Systems, 2015, pp. 1–9.
[39] N.B. Shah, D. Zhou, Y. Peres, Approval voting and incentives in crowdsourcing, in: Proceedings of International Conference on Machine Learning, 2015, pp. 10–19.
[40] A.D. Shaw, J.J. Horton, D.L. Chen, Designing incentives for inexpert human raters, in: Proceedings of the ACM Conference on Computer Supported Cooperative Work, 2011, pp. 275–284.
[41] R.M. Shiffrin, R.M. Nosofsky, Seven plus or minus two: a commentary on capacity limitations, Psychol. Rev. 101 (2) (1994) 61–357.
[42] C. Terwiesch, Y. Xu, Innovation contests, open innovation, and multiagent problem solving, Manage. Sci. 54 (9) (2008) 1529–1543.
[43] M. Venanzi, J. Guiver, G. Kazai, P. Kohli, M. Shokouhi, Community-based Bayesian aggregation models for crowdsourcing, in: Proceedings of International Conference on World Wide Web, 2014, pp. 155–164.
[44] J. Vuurens, A.P. de Vries, C. Eickhoff, How much spam can you take? an analysis of crowdsourcing results to increase accuracy, in: Proceedings of ACM SIGIR Workshop on Crowdsourcing for Information Retrieval, 2011, pp. 21–26.
[45] H. Xie, J.C.S. Lui, J.W. Jiang, W. Chen, Incentive mechanism and protocol design for crowdsourcing systems, in: Communication, Control, and Computing, 2014, pp. 140–147.
[46] M. Yin, Y. Chen, Bonus or not? learn to reward in crowdsourcing., in: Proceedings of International Joint Conferences on Artificial Intelligence Organization, 2015, pp. 201–208.
[47] M. Yin, Y. Chen, Y.-A. Sun, The effects of performance-contingent financial incentives in online labor markets., in: Proceedings of AAAI, 2013.
[48] M. Yin, Y. Chen, Y.-A. Sun, Monetary interventions in crowdsourcing task switching, in: Second AAAI Conference on Human Computation and Crowdsourcing, 2014.
[49] H. Zhai, T. Lingren, L. Deleger, Q. Li, M. Kaiser, L. Stoutenborough, I. Solti, Web 2.0-based crowdsourcing for high-quality gold standard development in clinical natural language processing, J. Med. Internet Res. 15 (4) (2013).
[50] J. Zhang, V.S. Sheng, Q. Li, J. Wu, X. Wu, Consensus algorithms for biased labeling in crowdsourcing, Inf. Sci. 382 (2017) 254–273.