Computing and Informatics, Vol. 37, 2018, 969–991, doi: 10.4149/cai_2018_4_969

ANNOTATING WEB TABLES WITH THE CROWD

Ning Wang, Huaxi Liu

School of Computer and Information Technology, Beijing Jiaotong University
No. 3 Shangyuancun, Haidian District, 100044 Beijing, China
e-mail: {nwang, 13120407}@bjtu.edu.cn

Abstract. The Web contains a large amount of structured tables, most of which lacks header rows. Algorithmic approaches have been proposed to recover semantics for web tables by annotating column labels and identifying subject columns. However, state-of-the-art technology is not yet able to provide satisfactory accuracy and recall. In this paper, we present a hybrid machine-crowdsourcing framework that leverages human intelligence to improve the performance of web table annotation. In this framework, machine-based algorithms are used to prompt human workers with candidate lists of concepts, while an improved K-means algorithm based on novel integrative distance is proposed to minimize the number of tuples posed to the crowd. In order to recommend the most related tasks for human workers and determine the final answers more accurately, an evaluation mechanism is also implemented based on Answer Credibility which measures the probability of a worker's intuitive answer being the final answer for a task. The results of extensive experiments conducted on real-world datasets show that our framework can significantly improve annotation accuracy and time efficiency for web tables, and our task reduction and answer evaluation mechanism is effective and efficient for improving answer quality.

Keywords: Crowdsourcing, semantic recovery, web tables, information integration

1 INTRODUCTION

Structured information of tables has a great value. In Google's recent research [1], the table schema and subject column are used to find related tables and to integrate multiple tables. Column labels are also used to learn binary relationships between multiple columns and class labels on cells [2]. The World Wide Web consists of a huge number of structured data in the form of HTML tables, most of which lacks header rows [3]. Some researchers tried to recover semantics of web tables by using large knowledge bases [4, 5], but the results are far from being perfect. According to the experimental results of [4], the precision of finding top-k concepts for each column using existing work is often low.

Recent studies have shown that crowdsourcing could be used effectively to solve problems that are difficult for computers, such as travel planning [6], open querying [7] and schema matching [8]. This can be applied for semantic recovery of web tables as well. For example, let us look at a simple table from a web page presented in Table 1 a). Human workers can easily decide that the concept for the third column is *capital* while the algorithm based on Probase [9] will return a candidate concept list *(city, large city, big city, capital, ...)*. For the second column of the table in Table 1 b), since the values are all numeric, it is really hard for the system to decide the concept for this column. But a human worker often can accurately determine that the concept is *price* according to the other columns' values. In addition to column label annotation, a worker is also able to play an important role in subject column identification.

1	2	3
America	English	Washington
China	Chinese	Beijing
Japan	Japanese	Tokyo

a) Example I

1	2	3
iPhone 4S 8 GB	2 217	white
iPhone 5C 8GB	3 159	black
Samsung S7572	818	white

b) Example II

Table 1. Web table

As we can see from the above analysis, it is difficult to provide satisfactory accuracy and recall when annotating web tables only by computer algorithms. To tackle this problem, we propose a hybrid machine-crowdsouring framework, which combines the strength from computer algorithms with human intelligence to find the best answers for web table annotation.

It is a non-trivial task to develop a framework to annotate web tables with crowdsourcing. First, if a table contains too many tuples, it is a really boring task for a worker to browse the whole table and decide its headers and subject column. Second, a worker may wonder how to start if they could not get any prompt of candidate information from the system. Third, when a worker lacks the required knowledge for handling a complex job, the human contributed results can be arbitrarily bad.

In this paper, we present a hybrid machine-crowdsourcing framework that leverages human intelligence to improve the performance of web table annotation. To the best of our knowledge, we are the first to leverage crowdsourcing for recovering semantics of web tables. To achieve that, our framework starts with prompting human workers with candidate lists of concepts provided by machine-based algorithms to help the workers to annotate column labels and identify subject keys. Then the workers are presented with a small number of representative tuples in the table by clustering. If necessary, a worker could see more tuples which are similar with the representative one. Finally, an evaluation mechanism is implemented based on Answer Credibility according to the setting of expertise and practical performance in order to recommend the most related tasks for workers and decide the final answers.

We summarize our contributions below:

1. We propose a hybrid machine-crowdsourcing framework for web table semantic recovery problem.
2. We present crowd with a small number of representative tuples in the table for task reduction by clustering similar tuples based on integrative distance biased towards significant attributes.
3. We propose Answer Credibility to evaluate the probability of a human worker's intuitive answer being the final answer for a task.
4. We establish an evaluation mechanism based on Answer Credibility, which is used to recommend related tasks and determine the final answers for each task.
5. We have conducted extensive experiments based on real web tables and hundreds of crowdsourcing tasks, which demonstrate that our framework can significantly improve annotation accuracy and time efficiency for web tables, and our task reduction and answer evaluation mechanism is effective and efficient for improving answer quality.

2 ANNOTATING WEB TABLES BASED ON PROBASE

Probase uses the world as its model which has an extremely large concept/category space (2.7 million categories) harnessed from billions of web tables and many years worth of searching logs [9]. As those concepts are automatically acquired from web pages authored by millions of users, it is probably true that they cover most concepts in our mental world (about worldly facts). In addition, it has a large data space, a large attribute space and a large relationship space. These characteristics make it perfect for getting candidate labels and entity columns of web tables.

To help human workers to annotate column labels and identify subject keys, it is necessary to prompt them with candidate lists of concepts by machine-based algorithms.

We leverage Probase to find concepts by column in Probase which share at least one cell value with that in the column of the web table, and estimate the probability of a concept c_k given a set of instances $E = \{e_1, \dots, e_n\}$ using a naive Bayes model as follows.

$$P(c_k|E) = \frac{P(E|c_k)P(c_k)}{P(E)} \propto P(c_k)\prod_{j=1}^{N} P(e_j|c_k), \tag{1}$$

$$P(e_i|c_k) = \frac{n(e_i, c_k)}{n(c_k)} \tag{2}$$

where $n(e_i, c_k)$ is the occurrence frequency of pair (e_i, c_k), and $n(c_k)$ is the occurrence frequency of c_k. Then the concepts are sorted by the probability and we choose the top-k concepts as candidate headers of the column.

Finally, we take the evidence-based method introduced in [5] to detect the candidate subject column in a web table.

3 REDUCING UNCERTAINTY OF ANNOTATION BY CROWDSOURCING

3.1 Task Reduction by Clustering Similar Tuples

When human workers are asked to complete tasks by browsing the whole tables, they will not only spend much time on tasks but also generate poor quality answers. Therefore,we propose a novel solution for task reduction, that is to cluster similar tuples and present workers with representative ones.

3.1.1 Integrative Distance

A web table is composed of columns with various data types. In fact, some of columns are more significant for clustering. In rough set, the core attribute is thought to have a greater contribution to classification and decision. To make the calculation of distance between two tuples effective, more weight should be put on significant attributes during clustering which are either core attributes [10] or representative attributes.

For the i^{th} column in a web table T, if we get top-k candidate concept set $CH = \{ch_1, \dots, ch_k\}$ with the probability set $P = \{p_1, \dots, p_k\}$ for this column, and top-k candidate concept set $RC = \{rc_1, \dots, rc_k\}$ for whole table T based on Probase, the representative possibility of the i^{th} column, which describes the relevance degree between this column and table T, is computed as $rp_i = \frac{\sum_{c_j \in CH \cap RC} p_j}{|CH \cap RC|}$, where $p_j \in P$ is the probability of candidate concept c_j for this column. Representative attributes of T are those attributes with representative possibility which are beyond a threshold t_r.

Definition 1 (Significant Attribute)**.** Assume $CA = \{CA_1, \dots, CA_m\}$ denotes the core attribute set of a web table and $RA = \{RA_1, \dots, RA_n\}$ denotes its representative attribute set, then each one in their union set $SA = CA \cup RA$ is a significant attribute.

We use integrative distance biased towards significant attributes to evaluate the similarity between tuples in a web table.

Definition 2 (Integrative Distance)**.** For a web table T with attribute set $A = \{a_1, \dots, a_n\}$, given the significant attribute set $SA = \{sa_1, \dots, sa_l\}$, the distance set $D_{ij} = \{d_{ij}^1, \dots, d_{ij}^n\}$ between two tuples t_i and t_j in table T, where d_{ij}^k denotes distance between corresponding $a_k(k = 1, \dots, n)$ in t_i and t_j, and the weight set

$$w_{ij} = \begin{cases} \left\{ w_{ij}^1, \dots, w_{ij}^n \middle| w_{ij}^p{}_{1 \le p \le n, a_p \in SA} > w_{ij}^q{}_{1 \le q \le n, a_q \notin SA} \right\}, & SA \neq \phi, \\ \left\{ w_{ij}^1, \dots, w_{ij}^n \middle| w_{ij}^p{}_{1 \le p \le n} = w_{ij}^q{}_{1 \le q \le n} \right\}, & SA = \phi. \end{cases} \tag{3}$$

assigned on D, the integrative distance between t_i and t_j is calculated as follows.

$$D_{ij} = \sum_{k=1}^{n} w_{ij}^k d_{ij}^k. \tag{4}$$

Integrative distance is proposed to combine Euclidean distance with Jaccard similarity on different attributes in a web table. The distance function is biased towards the existing significant attributes.

3.1.2 Clustering Similar Tuples Based on Integrative Distance

In order to reduce the number of tuples posed to the crowd, we design a clustering algorithm named Clustering Algorithm based on Integrative Distance (CAID), which is an improved K-means algorithm, to cluster similar tuples in a web table and prompt workers with representative ones that are nearest to each of cluster centers.

We select K-means instead of K-medoids for computational constraints in crowdsourcing environment. Traditional K-means is improved in CrowdSR to adapt to web tables by using the integrative distance biased towards significant attributes.

As shown in Algorithm 1, CAID is different from naive K-means in step 3 and step 6, which are also our improvements. In step 3, we firstly get the signficant attributes of a table based on Probase. From step 5 to step 8, we get the initial clustering centers by finding the tuples which are farthest to each other. In step 6, we find the next tuple for initial clustering centers by evaluating the integrative distance biased towards significant attributes between tuples. Integrative distance is also used to reassign tuples and update clustering centers in step 10 and step 12.

3.2 Improving Answer Quality Based on Answer Credibility

Quality control is very important for a crowdsourcing platform. Based on the fact that a worker can give high quality answers when he does tasks with expertise, we propose Answer Credibility to evaluate his acquaintance with fields assigned by a task. In order to improve answer quality, we establish an evaluation mechanism based on Answer Credibility to recommend related tasks and decide the final answers.

Algorithm 1 CAID Clustering Algorithm

```
procedure getCluserResult($T, K$)
    $F \leftarrow True$
    $SA \leftarrow getSignificantAttributes(T)$
    $addOneRandTuple(T, R)$
    while $R.size() <= K$ do
        $NT \leftarrow getNextFarthestTuple(T)$
        $addTuple(R, NT)$
    end while
    while $F$ do
        $reassignTuple(T, R)$
        for all $S \in R$ do
            $replaceCentroidWithMean(S, R)$
        end for
        $F \leftarrow centroidChange(R)$
    end while
    return $R$
end procedure
```

3.2.1 Answer Credibility

In our framework, users are divided into requesters who would like to publish tasks, and workers who are willing to accept and complete tasks. We use the set $F = \{f_1, \ldots, f_m\}$ to describe the whole set of fields and each task is assigned with several fields in F by the requester when published.

For each worker, his answer credibility for a task is based on the degree of his acquaintance with each related field. Field credibility is therefore proposed to evaluate a worker's knowledge about some field in F, which is evaluated by following aspects.

1. Settings of expertise: At the beginning, each worker is required to choose several fields from F as his expertise. The initial score of each field is set by our system and the score of expertise (HS) is higher than the score of other fields (LS). Let $E = \{e_1, \ldots, e_v | e_i \in F, 1 \leq i \leq v\}$ denote the set of expertise, the elementary score set for a worker is

$$ES = \left\{ es_1, \ldots, es_m \;\middle|\; \sum_{1 \leq i \leq m} es_i = 1 \right\} \tag{5}$$

where

$$es_i = \begin{cases} HS, & f_i \in E, \\ LS, & f_i \notin E, \end{cases} \quad (0 < LS < HS < 1). \tag{6}$$

2. Brilliance test: Furthermore, a new user is required to join our brilliance test to check his actual field credibility when he registers as a worker. Assume a worker completes M test tasks. Let $F_M = \{tf_1, \ldots, tf_M\}$ denote the field set selected by the worker and corresponding score set of test tasks be $S_M = \{s_1, \ldots, s_M\}$. Brilliance test score set is

$$BS = \{bs_1, \ldots, bs_m\} \tag{7}$$

where

$$bs_i = \begin{cases} \frac{s_j}{\sum_{1\le k\le M} s_k}, & (f_i \in F_M) \wedge (f_i = tf_j), \\ 0, & (f_i \notin F_M) \vee ((f_i \in F_M) \wedge (f_i \neq tf_j)). \end{cases} \tag{8}$$

3. Actual performance of doing tasks: For a task T, let $F_T = \{f_1, \ldots, f_t\}$ denote the field set assigned on T by requester, $IAns = \{Ians_1, \ldots, Ians_n, Ians_{n+1}\}$ denote a worker's intuitive answers for n column labels and one subject column, $FAns = \{Fans_1, \ldots, Fans_n, Fans_{n+1}\}$ denote the final answer summarized from several workers' intuitive answers for T, the incremental performance score set is

$$\Delta PS = \{\Delta ps_1, \ldots, \Delta ps_m\} \tag{9}$$

where

$$\Delta ps_i = \begin{cases} \frac{\sum_{1\le k\le n+1} |IAns(k)=FAns(k)|}{|IAns|}, & f_i \in F_T, \\ 0, & f_i \notin F_T. \end{cases} \tag{10}$$

Then the actual performance score set for this worker is

$$PS = \{ps_1^k, \ldots, ps_m^k\} \tag{11}$$

where

$$ps_i^k = \begin{cases} 0, & k = 0, \\ \frac{ps_i^{k-1} + \Delta ps_i}{\sum_{1\le j\le m} (ps_j^{k-1} + \Delta ps_j)}, & k \ge 1. \end{cases} \tag{12}$$

For a worker U and field $f_i \in F$, if his elementary score, brilliance test score and performance score for field f_i are es_i, bs_i and ps_i, the field credibility of U for f_i could be computed as

$$fc_i = \begin{cases} \lambda es_i + (1-\lambda) bs_i, & ps_i = 0, \\ \lambda_1 ps_i + \lambda_2 bs_i + \lambda_3 es_i, & ps_i > 0 \end{cases} \tag{13}$$

where $(0 < \lambda < 1) \wedge \left(\sum_{1\le j\le 3} \lambda_j = 1\right)$ and they are weights for corresponding possibility. For a new worker, we only take the settings of expertise and brilliance test into consideration, and he could improve his credibility only by doing actual tasks.

Definition 3 (Answer Credibility)**.** For a worker U working on task T, given the field set $F_T = \{f_1, \dots, f_t\}$ assigned by requester on T, field credibility set $FC = \{fc_1, \dots, fc_t\}$ and intuitive answers $IAns = \{Ians_1, \dots, Ians_n, Ians_{n+1}\}$ of U on task T. The answer credibility of U for T is computed as

$$AC = \sum_{i=1}^{t} fc_i. \tag{14}$$

Answer credibility is modeled to evaluate the probability of which a worker's intuitive answer comes to be the final answer for a task, which could be used to recommend tasks for workers who are more competent for and to decide the final answers.

3.2.2 Evaluation Mechanism Based on Answer Credibility

Task Recommendation: Let $T = \{T_1, \dots, T_n\}$ denote the task list. For a worker U, we get his answer credibility set $AC = \{AC_1, \dots, AC_n\}$ as described before. Then we recommend $T_{rec} = \{T_1, \dots, T_k\}$ for him with top-k answer credibility values.

Answer Decision: In order to get final answer from multiple workers, a voting mechanism is built on Answer Credibility. For instance, when u workers working on task T, $AC_T^1, \dots, AC_T^u$ are their answer credibility values for T, $IAns^i (i = 1, \dots, n+1)$ is the set of the i^{th} answers for i^{th} column label $(i = 1, \dots, n)$ or subject column $(i = n + 1)$ gathered from u workers. If we use U^i to denote the set of workers whose candidate answers in $IAns^i$ are the same, the score of this candidate answer in $IAns^i$ is evaluated as $\sum_{u \in U^i} AC_T^u$, and the final answer for $IAns^i$ is the candidate answer with the maximum score.

4 SYSTEM OVERVIEW

We have implemented CrowdSR, a crowd enabled system for semantic recovering of web tables based on a hybrid machine-crowdsourcing framework [11]. CrowdSR is implemented in JSP with SQL Server database as back-end. Figure 1 depicts the system architecture.

User Interface is used to interact with users (both requesting a task and contributing to it). Basically, *DB* stores answers collected from the crowd, targeting information and details about each user and task. *Task builder* receives requests from requesters and builds tasks. *Crowd Manager* constantly receives an updated list of online workers from targeting information in *DB*.

Semantic Recovery Component is responsible for finding candidate column labels and subject columns for each task based on the *Semantic Library* (we now use Probase, which could be replaced by other third-party libraries easily). When a task is published, it is enriched with candidate answers by the *Semantic Recovery Component*.

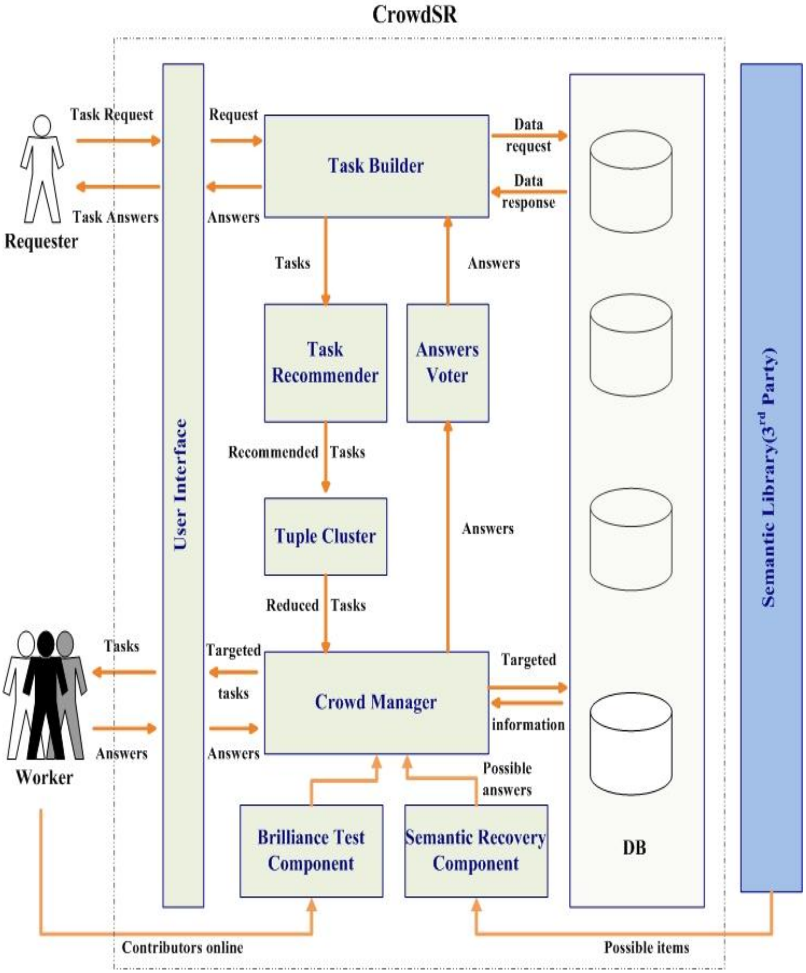


Figure 1. CrowdSR architecture

At the beginning, a worker is required to join our brilliance test to evaluate his understanding of each field through the *Brilliance Test Component*. *Task Recommender* and *Answers Voter* implement our evaluation mechanism based on Answer Credibility. Once a worker logs in, the *Task Recommender* recommends task list for him based on his answer credibility. While the deadline of a task arrives, received answers are passed to *Answers Voter* to decide the final answer.

Tuple Cluster executes our CAID algorithm to cluster similar tuples for each task. Workers complete tasks via a question-choice game where they are shown representative values of each column, along with a set of candidate table headers and subject column, and are asked to select ones most matched. Furthermore, a worker could check the alteration of his answer credibility based on his performance.

5 EXPERIMENTS AND RESULTS

The goals of our experiments are:

1. to evaluate the effectiveness of our hybrid machine-crowdsourcing framework for web table annotation based on Probase knowledge base;
2. to validate the performance of our CAID clustering methods based on integrative distance;
3. to evaluate the effectiveness of our evaluation mechanism based on Answer Credibility for task recommendation and answer decision.

5.1 Experiment Setup

Google Fusion Table (GFT) is a popular web application provided by Google to allow people, including those with no database expertise, to manage their data [13]. We choose GFT as one of our source of data for the reason that GFT tables have a neat, regular structure and the Australian government has already published its database on GFT. Besides, we extract large-scale web tables named WT which covers kinds of fields for crowdsourcing tasks to evaluate our mechanism based on Answer Credibility. Headers of tables in our dataset are known, but we remove them during experiment. Table 2 gives statistics of our data sets.

Dataset	Size (MB)	# Columns	# Rows	# Cells
GFT	4.5	261	15 132	3 949 452
WT	5.6	479	9 290	4 449 910

Table 2. Experimental data set

To conduct the experiments, we have implemented CrowdSR in JSP while using SQL Server as the database engine. The architecture of CrowdSR is presented in Figure 1. Compared with some existing platforms, such as Amazon Mechanical Turk (AMT) [22] and CrowdFlower [23], CrowdSR implemented task reduction and answer quality control. All our experiments were conducted on an Intel® Core$^{\text{TM}}$ 2.13 GHz computer with 2 GB of RAM running Windows 7.

5.2 Evaluation on Performance of CAID Clustering Algorithm

We select 10 representative web tables from different fields in GFT, which consist of labels used to get the clustering precision, to evaluate the clustering performance of our CAID clustering algorithm. Table 3 gives statistics of those tables. We compare CAID with naive K-means from two aspects, which are clustering precision and

Table Name	Size (KB)	# Columns	# Rows	# Cells
Agriculture	384	3	2 697	8 091
Architecture	71.5	6	347	2 081
Privacy Survey	1 028	45	1 000	45 000
Disaster	324	9	373	3 357
School	64.5	13	132	1 716
Crime	229	8	565	4 520
Books	303	9	786	7 074
Fish	154	7	452	3 164
Social Statistics	57	5	35	175
Wetland	584	7	1 000	7 000

Table 3. Statistics of 10 representative tables in GFT

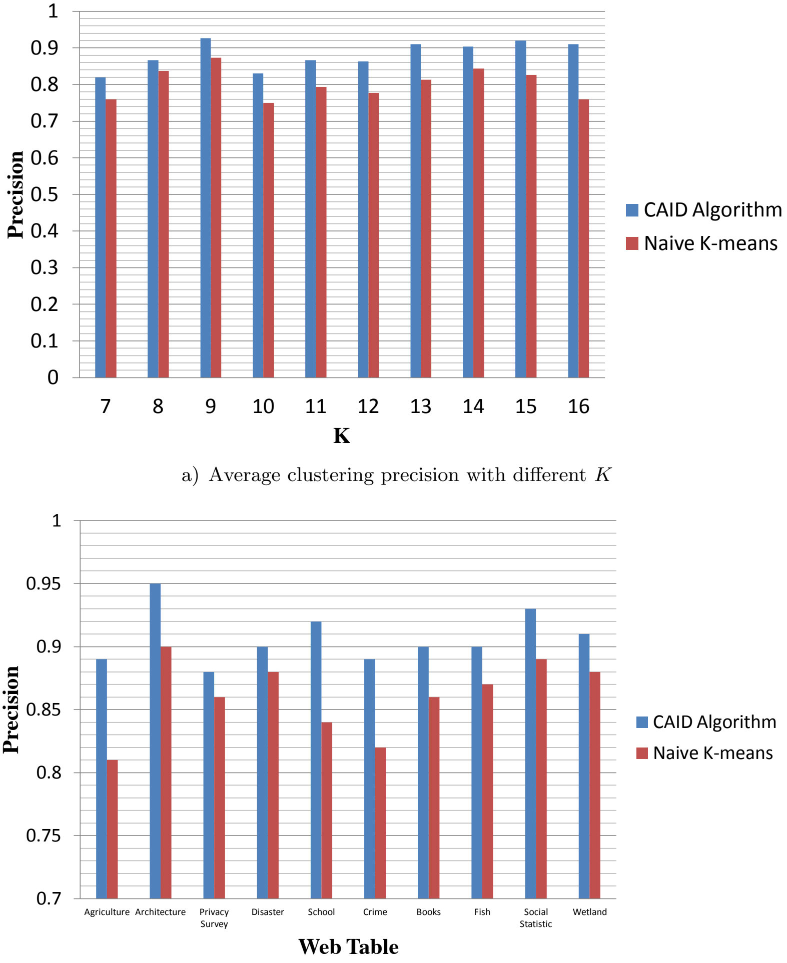
time efficiency. As the precision of naive K-means depends greatly on the selection of initial clustering centers, we ran it for 500 times to get its average precision just for fairness.

Before the experiment, we also took two factors below into consideration.

1. The number of clustering centers K: The size of K has influence on clustering algorithm. In order to compare the performance, we apply naive K-means and CAID with different size of K between $[7, 16]$. We select the region for experiments because it is easy for workers to finish tasks by browsing a small number of tuples.
2. Distance between tuples: As described before, CAID uses integrative distance to calculate similarity between tuples, which combines Euclidean distance with Jaccard similarity and gives high weight on significant attributes. Since a web table is usually composed of columns with various data types, we also combine Euclidean distance with Jaccard similarity for naive K-means just for fairness, but the weight of each column is equal.

Figure 2 a) shows the average precision of two algorithms with different k value in $[7, 16]$. The precision and time cost of two algorithms on 10 representative tables with $K = 9$ are displayed in Figures 2 b) and 2 c), respectively. We have following observations:

1. The precision of CAID is obviously higher than that of naive K-means, either average precision on different tables or precision on single table, because of the improvement of initial clustering centers and use of integrative distance. The precision of two methods varies with the change of k value, but both of them achieve the best with $K = 9$.
2. The time cost of naive K-means is much lower than that of CAID. Compared with naive K-means, CAID spends much time getting the initial clustering centers and significant attributes for each web table as described in Algorithm 1. The time for table of privacy survey and wetland are obviously higher as they



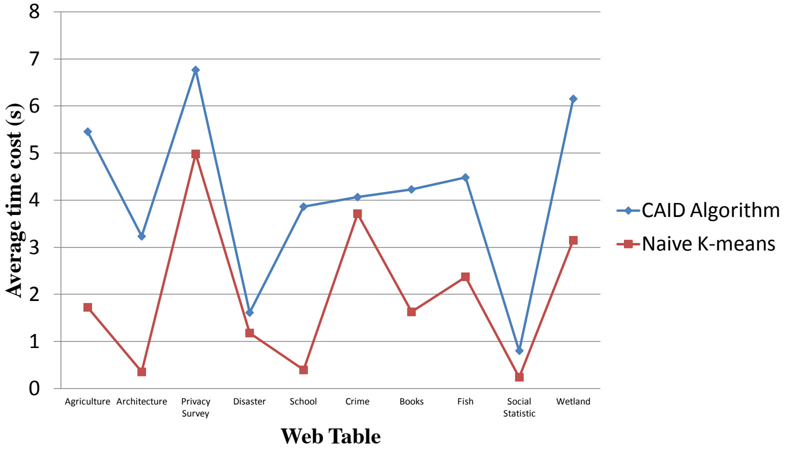
a) Average clustering precision with different K

b) Clustering precision with $K = 9$

contain much more tuples. We will build index and improve time efficiency of CAID in future work.

5.3 Effectiveness of Hybrid Framework

We select web tables from WT data set which covers 12 different fields to publish crowdsourcing tasks. Tables from WT have original headers, and we labeled entity columns for each table as the golden standard. We enlisted hundreds of students from our university to join our experiments as workers. Each task is assigned with three fields when published and one task could be completed by several workers.



c) Clustering time cost with $K = 9$

Figure 2. Performance of CAID algorithm

Before doing the tasks, each worker was required to set his expertise and join our brilliance test. When he signed in, he was required to complete four kinds of tasks as following:

1. Recommended task with Representative tuples (RR): The task was recommended by the evaluation mechanism in CrowdSR and the workers were shown representative tuples in this task at first. They could also see more similar tuples by clicking corresponding buttons.
2. Recommended task with Whole tuples (RW): The task was recommended but the workers were shown all tuples in a table even if it contains hundreds of tuples.
3. General task with Representative tuples (GR): The general task that may not be recommended by our system and the workers were shown representative tuples in this task.
4. General task with Whole tuples (GW): The general task that may not be recommended by our system and the workers were shown all tuples in this task.

We use abbreviations RR, RW, GR and GW to refer the tasks in the following section. Finally, we collected the candidate answers and the time cost of each task to evaluate the effectiveness of our hybrid framework.

5.3.1 Evaluation on Hybrid Machine-Crowdsourcing Method

To compare the effectiveness of machine-based method with our hybrid machine-crowdsourcing method, we use precision, recall and F-measure, defined in Equations (15), (16) and (17):

$$Precision = \frac{|CAL|}{|AL|}, \tag{15}$$

$$Recall = \frac{|AL|}{|WL|}, \tag{16}$$

$$F = \frac{(1+\alpha) \times Precision \times Recall}{\alpha \times Precision + Recall}. \tag{17}$$

Precision measures the percentage of annotated headers or entity columns which are correct, and recall measures the percentage of headers or entity columns which could be annotated. F-measure is the weighted harmonic mean of precision and recall in which precision and recall are evenly weighted if $\alpha = 1$, and we set $\alpha = 1$ in this paper.

In Equations (15) and (16), WL is the set of whole headers or entity columns, AL is the set of annotated headers or entity columns and CAL is the set of headers or entity columns which are correctly annotated.

We took following two issues into consideration when collecting experimental data.

1. Answer Decision: Our hybrid machine-crowdsourcing framework can decide the final answers by the voting mechanism based on Answer Credibility. However, we did a favor for machine-based method. We chose the top three candidate concepts and the top one entity column returned by algorithm as the final answer set, and it is thought to be correctly annotated if one of concepts in answer set matched with golden standard.
2. Synonyms Principle: WT data set has original headers and labeled entity columns as the golden standard to judge the accuracy of results. But it is inevitable to face the synonyms problem for table headers. For example, we could get *author*, *writer* as the candidate headers for a column. So we made a principle that the word which is synonymous with the golden standard turns to be right. We use WordNet [24] as the synonymous words library to solve the problem.

According to the original experimental data in Table 4, the annotation precision, recall and F-measure on two approaches for headers and entity columns are calculated and shown in Figure 3. We could have the following observations:

1. For machine-based method, the precision of headers and entity columns are only 41.1 % and 41.38 %. But the average precision of our hybrid method are 57.29 % and 81.23 %. Our hybrid method could obviously improve the precision of web table annotation, especially for entity columns. The growth rate of the average precision on our hybrid approach for column headers (16.19 %) is much lower than that for entity columns (39.83 %) because the machine-based method uses the top three candidates rather than just the single top one candidate as final answers for headers.

Method	Machine-based	Hybrid Machine-crowdsourcing			
		RR	RW	GR	GW
# Total tables	130	41	51	66	55
# Total columns	479	249	246	275	301
# Total rows	9 290	1 746	2 048	3 997	4 810
# Annotated columns	163	155	179	186	182
# Correctly annotated columns	67	108	118	79	93
# Annotated entity columns	29	36	48	49	50
# Correctly annotated entity columns	12	31	40	37	40

Table 4. Original experimental data

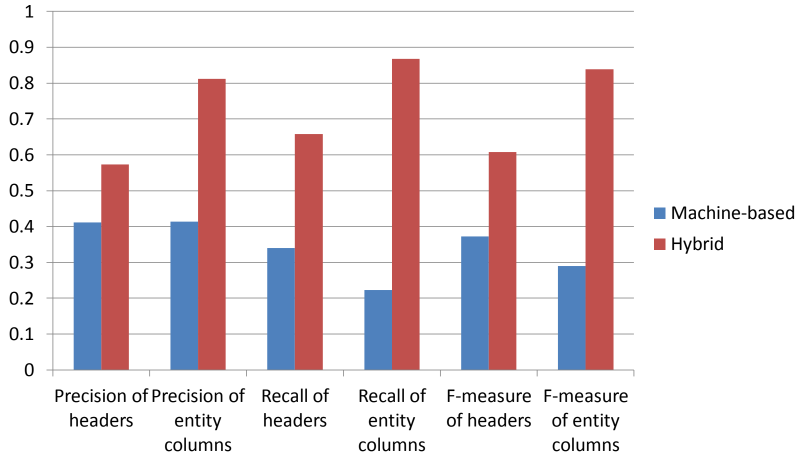


Figure 3. Comparison of annotation performance on two approaches

2. The machine-based algorithm totally processed 130 web tables with 479 columns and returned 29 candidate entity columns and 163 table headers. The recall of table headers and entity columns are only 163/479 = 34.0 % and 29/130 = 22.3 %. We gathered results from over five hundred crowdsourcing tasks. As one task could be completed by several workers, our hybrid method processed more than 1 000 columns. The average percentage of annotated columns and entity column are 74.88 % and 85.9 %, which are significantly higher than a machine-based method.

3. As the precision and recall are higher, the F-measure of our hybrid method is also better than that of a machine-based method.

It is difficult for computer algorithm to annotate web tables mainly due to two reasons. At first, the data format of web tables is irregular and there is some noisy data. Secondly, the web tables usually consist of long text and it is really hard for computer algorithm to analyze their semantics. Our hybrid method could significantly improve the performance of web table annotation by using crowd wisdom.

5.3.2 Benefits from Task Reduction

As described before, human workers are required to complete four kinds of tasks, which are Recommended task with Representative tuples (RR), Recommended task with Whole tuples (RW), General task with Representative tuples (GR) and General task with Whole tuples (GW). Task reduction means providing reduced tasks with representative tuples, such as RR or GR tasks, in which workers were firstly shown representative tuples and they could also see more similar data with current tuples when necessary. On the contrary, a table is completely shown in a page when workers are doing RW and GW tasks which are named full tasks. Although most of web tables consists of hundred of tuples and even some of tables contain thousands of tuples, the workers doing full tasks need to browse the whole table to decide their answers. We evaluate the benefits from task reduction by comparing the performance of reduced tasks with that of full tasks.

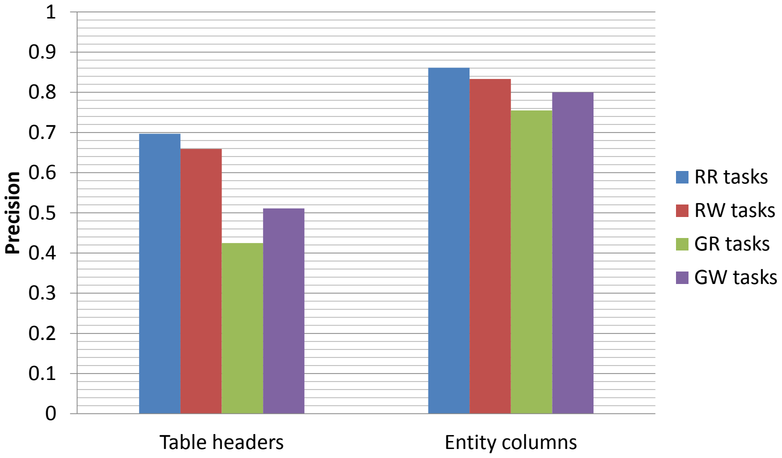
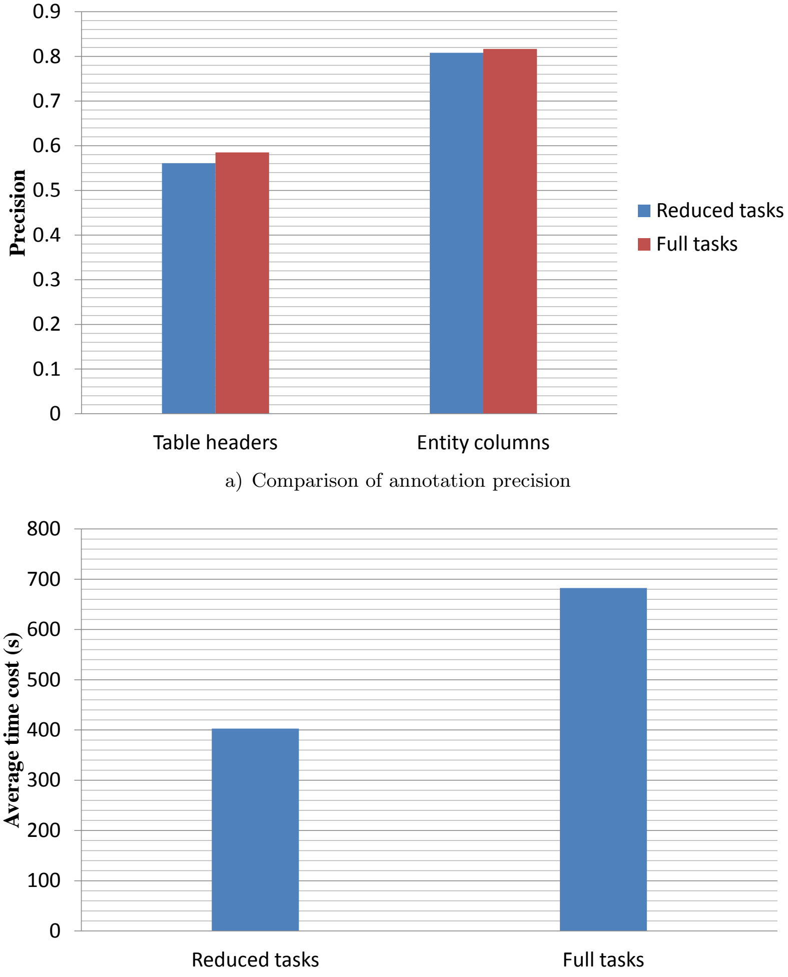


Figure 4. Annotation precision of RR, RW, GR and GW tasks

Figure 4 gives the annotation precision of RR, RW, GR and GW tasks. The average precision of reduced tasks and full tasks, and the corresponding time cost are shown in Figures 5 a) and 5 b), respectively. We have the observations as follows:

1. As shown in Figures 4 and 5 a), although the precision of RR tasks is better than that of RW tasks, the average precision of full tasks is yet a little higher than that of reduced tasks mainly because the precision of GR tasks is much lower than that of GW tasks. For general tasks with the fields workers may not be familiar with, it may be better to browse whole tuples to make decision. In



a) Comparison of annotation precision

b) Comparison of time cost

Figure 5. Comparison between reduced and full tasks

CrowdSR, we have an effective evaluation mechanism based on Answer Credibility to recommend tasks for workers who are more competent for, which is beneficial for improving annotation precision.

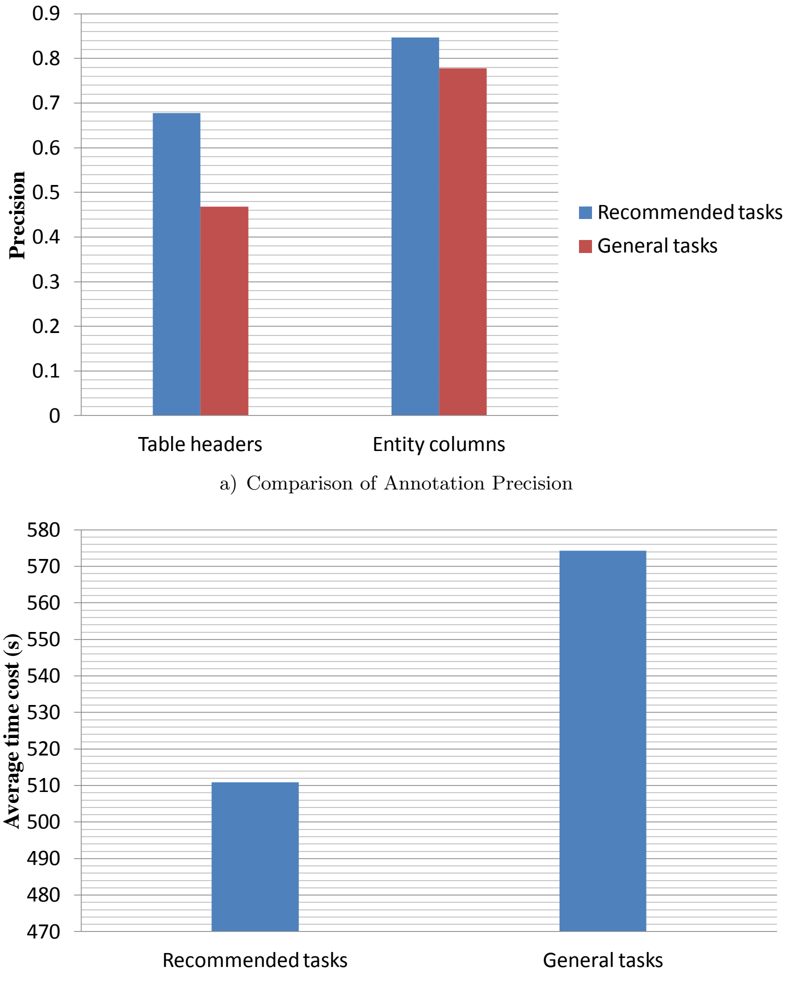
2. As in Figure 5 b), the average time cost of reduced tasks is only 402.62 seconds, which is nearly 70.0 % lower than that of full tasks. The result demonstrates that task reduction could obviously reduce the time cost.

Furthermore, in order to investigate the workers' attitude towards different kinds of tasks, we conducted a questionnaire survey after they complete the tasks. 83.3 %

of them thought the reduced tasks are enjoyable, while 66.7 % of them considered the full tasks are really boring. The results prove that human workers prefer to do reduced tasks rather than full tasks.

5.3.3 Effectiveness of Answer Evaluation Mechanism

In this section, we will validate the effectiveness of answer evaluation mechanism by comparing the performance of recommended tasks and general tasks.



a) Comparison of Annotation Precision

b) Comparison of Time Cost

Figure 6. Comparison between Recommended and General Tasks

CrowdSR recommends tasks for each worker according to his answer credibility. At the beginning, we get the initial answer credibility of each worker according to his expertise settings and performance of brilliance test. Their answer credibility values are dynamically changed when they are doing actual tasks.

As shown in Figures 6 a) and 6 b), the performance of recommended tasks is obviously better than that of general tasks. The average annotation precision of recommended tasks increased about 14.0 % over that of general tasks, also with lower time cost. The result demonstrates that our answer evaluation mechanism works much effectively.

6 RELATED WORK

Semantic recovery for web tables is obviously important for web search to take advantage of those high quality sources of relational information. Deng et al. tried to determine the column concepts for web tables by using large knowledge bases [4] and Wang et al. used a method based on serveral kinds of evidence to find the entity column [5]. In addition, Venetis et al. tried to recover the semantics of web tables by enriching the table with additional annotations [12]. But according to their experimental results, the accuracy is far from perfect and the machine-based method is unstable [4, 5].

Using crowd wisdom for settling problems that computers fail to solve has attracted much research attention in recent years, especially in database and data mining communities. CrowdDB used human input via crowdsourcing to process queries that neither database systems nor search engines could adequately answer [7]. Amsterdamer et al. proposed the identification of frequent itemsets in human knowledge domains by posing questions to the crowd [13]. A novel algorithm is developed in CrowdPlanr for efficiently harnessing the crowd to assist in answering planning queries [6]. Besides, crowdsourcing-based solutions of many complex algorithms are also developed, such as crowd-assisted search problem in a graph [15], crowd-based entity resolution for data cleaning [16, 17, 18, 19] and schema matching via crowdsourcing for data integration [8]. In this paper, we present a hybrid machine-crowdsourcing framework that leverages human intelligence to improve the performance of web table annotation.

There also exist several crowdsourcing platforms, such as AMT [22] and CrowdFlower [23], but they are general platforms which are facing the problem that humans are prone to errors and an incorrect answer may be provided especially when a person is lacking the required knowledge for handling the task. In AMT, a job is split into many HITs (Human Intelligence Tasks) and each HIT is assigned to multiple workers so that replicated answers are obtained and the majority voting strategy is adopted for deciding the final answer. As human cost is expensive, we will have to pay a high cost if we assign each HIT to too many workers. In order to implement quality management on AMT, Ipeirotis et al. presented an algorithm that separated the unrecoverable error rate from bias by generating a scalar score for distinguishing workers who are careful but biased with those spammers [20]. Furthermore, Karger et al. introduced a model in which a requester has a set of homogeneous labeling tasks he must assign to workers who arrive online [21]. They proposed an assignment algorithm based on random graph generation and showed that their technique is order-optimal in terms of labeling budget. Although these are important aspects for a crowdsourcing platform, our focus here is different. We implement an evaluation mechanism based on Answer Credibility to improve answer quality for annotating web tables.

To the best of our knowledge, we are the first to leverage crowdsourcing for semantic recovering of web tables. Compared with all of them, we firstly propose to reduce tasks by using clustering algorithm, define answer credibility for answer quality control by task recommendation and answer decision.

7 CONCLUSION

In this paper, we present a hybrid machine-crowdsourcing framework that leverages human intelligence to improve the performance of web table annotation. We implement task reduction by minimizing the number of tuples posed to the crowd, task recommendation and answer decision by evaluating answer credibility of every worker for each task. Furthermore, we conduct extensive experiments based on real web tables and crowdsourcing tasks, which demonstrate that our framework can obviously improve annotation accuracy and time efficiency for web tables, and our task reduction and answer mechanism is effective and efficient for improving answer quality. In the future, we will try to improve the time efficiency of CAID algorithm and use crowdsourcing to solve table fusion problem.

Acknowledgement

This work is supported by the National Key R & D Program of China (2018YFC0809800) and the National Natural Science Foundation of China (61370060, 61300071).

REFERENCES

[1] Das Sarma, A.—Fang, L.—Gupta, N. et al.: Finding Related Tables. Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, 2012, pp. 817–828, doi: 10.1145/2213836.2213962.

[2] Limaye, G.—Sarawagi, S.—Chakrabarti, S.: Annotating and Searching Web Tables Using Entities, Types and Relationships. Proceedings of the VLDB Endowment, Vol. 3, 2010, No. 1-2, pp. 1338–1347, doi: 10.14778/1920841.1921005.

[3] Cafarella, M. J.—Halevy, A.—Wang, D. Z.—Wu, E.—Zhang, Y.: WebTables: Exploring the Power of Tables on the Web. Proceedings of the VLDB Endowment, Vol. 1, 2008, No. 1, pp. 538–549, doi: 10.14778/1453856.1453916.

[4] Deng, D.—Jiang, Y.—Li, G.—Li, J.—Yu, C.: Scalable Column Concept Determination for Web Tables Using Large Konwledge Bases. Proceedings of the VLDB Endowment, Vol. 6, 2013, No. 13, pp. 1606–1617.

[5] Wang, J.—Wang, H.—Wang, Z.—Zhu, K. Q.: Understanding Tables on the Web. In: Atzeni, P., Cheung, D., Ram, S. (Eds.): Conceptual Modeling (ER 2012). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7532, 2012, pp. 141–155.

[6] Lotosh, I.—Milo, T.—Novgorodov, S.: CrowdPlanr: Planning Made Easy with Crowd. Proceedings of the 2013 IEEE 29th International Conference on Data Engineering (ICDE), 2013, pp. 1344–1347, doi: 10.1109/ICDE.2013.6544940.

[7] Franklin, M. J.—Kossmann, D.—Kraska, T.—Ramesh, S.—Xin, R.: CrowdDB: Answering Queries with Crowdsourcing. Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data (SIGMOD '11), 2011, pp. 61–72, doi: 10.1145/1989323.1989331.

[8] Zhang, C. J.—Chen, L.—Jagadish, H. V.—Cao, C. C.: Reducing Uncertainty of Schema Matching via Crowdsourcing. Proceedings of the VLDB Endowment, Vol. 6, 2013, No. 9, pp. 757–768, doi: 10.14778/2536360.2536374.

[9] Wu, W.—Li, H.—Wang, H.—Zhu, K. Q.: Probase: A Probabilistic Taxonomy for Text Understanding. Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data (SIGMOD '12), 2012, pp. 481–492, doi: 10.1145/2213836.2213891.

[10] Han, S.-W.—Kim, J.-Y.: Rough Set-Based Decision Tree Using a Core Attribute. International Journal of Information Technology and Decision Making, Vol. 7, 2008, No. 2, pp. 275–290.

[11] Liu, H.—Wang, N.—Ren, X.: CrowdSR: A Crowd Enabled System for Semantic Recovering of Web Tables. In: Dong, X., Yu, X., Li, J., Sun, Y. (Eds.): Web-Age Information Management (WAIM 2015). Springer, Cham, Lecture Notes in Computer Science, Vol. 9098, 2015, pp. 581–583.

[12] Venetis, P.—Halevy, A.—Madhavan, J.—Paşca, M.—Shen, W.—Wu, F.—Miao, G.—Wu, C.: Recovering Semantics of Tables on the Web. Proceedings of the VLDB Endowment, Vol. 4, 2011, No. 9, pp. 528–538, doi: 10.14778/2002938.2002939.

[13] Amsterdamer, Y.—Grossman, Y.—Milo, T.—Senellart, P.: Crowd Mining. Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD '13), 2013, pp. 241–252, doi: 10.1145/2463676.2465318.

[14] Gonzales, H.—Halevy, A. Y.—Jensen, C. S. et al.: Google Fusion Tables: Web-Centered Data Management and Collaboration. Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD '10), 2010, pp. 1061–1066.

[15] Parameswaran, A.—Das Sarma, A.—Garcia-Molina, A.—Polyzotis, N.—Widom, J.: Human-Assisted Graph Search: It's Okay to Ask Questions. Proceedings of the VLDB Endowment, Vol. 4, 2011, No. 5, pp. 267–278, doi: 10.14778/1952376.1952377.

[16] Vesdapunt, N.—Bellare, K.—Dalvi, N.: Crowdsourcing Algorithms for Entity Resolution. Proceedings of the VLDB Endowment, Vol. 7, 2014, No. 12, pp. 1071–1082, doi: 10.14778/2732977.2732982.

[17] Wang, J.—Kraska, T.—Franklin, M. J.—Feng, J.: CrowdER: Crowdsourcing Entity Resolution. Proceedings of the VLDB Endowment, Vol. 5, 2012, No. 11, pp. 1483–1494.

[18] Wang, J.—Li, G.—Kraska, T.—Franklin, M.—Feng, J.: Leveraging Transitive Relations for Crowdsourced Joins. Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD '13), 2013, pp. 229–240.

[19] Whang, S. E.—Lofgren, P.—Garcia-Molina, H.: Question Selection for Crowd Entity Resolution. Proceedings of the VLDB Endowment, Vol. 6, 2013, No. 6, pp. 349–360.

[20] Ipeirotis, P. G.—Provost, F.—Wang, J.: Quality Management on Amazon Mechanical Turk. Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP '10), 2010, pp. 64–67, doi: 10.1145/1837885.1837906.

[21] Karger, D.—Oh, S.—Shah, D.: Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems. Operations Research, Vol. 62, 2014, No. 1, pp. 1–24, doi: 10.1287/opre.2013.1235.

[22] AMT Web Site. Available at: `https://www.mturk.com/mturk/welcome`.

[23] CrowdFlower Web Site. Available at: `http://www.crowdflower.com/`.

[24] WordNet Web Site. Available at: `http://wordnet.princeton.edu/wordnet/`.

Ning Wang received her Ph. D. degree in computer science in 1998 from Southeast University in Nanjing, China. She is currently serving as Professor in School of Computer and Information Technology, Beijing Jiaotong University, China. Her research interests include web data integration, big data management, data quality and crowdsourcing.

Huaxi Liu received his Master degree in computer science from Beijing Jiaotong University in 2016 and he currently works in Huawei Technologies Co., Ltd. His research interests include web data integration, data mining and crowdsourcing.