CrowdCleaner: Data Cleaning for Multi-version Data on the Web via Crowdsourcing

Yongxin Tong, Caleb Chen Cao, Chen Jason Zhang, Yatao Li, Lei Chen
Department of Computer Science and Engineering, Hong Kong University of Science & Technology, Hong Kong, China
{yxtong, caochen, czhangad, ylibg, leichen}@cse.ust.hk

***Abstract*— Multi-version data is often one of the most concerned information on the Web since this type of data is usually updated frequently. Even though there exist some Web information integration systems that try to maintain the latest update version, the maintained multi-version data usually includes inaccurate and invalid information due to the data integration or update delay errors. In this demo, we present *CrowdCleaner*, a smart data cleaning system for cleaning multi-version data on the Web, which utilizes crowdsourcing-based approaches for detecting and repairing errors that usually cannot be solved by traditional data integration and cleaning techniques. In particular, *CrowdCleaner* blends *active* and *passive* crowdsourcing methods together for rectifying errors for multi-version data.**

We demonstrate the following four facilities provided by *CrowdCleaner*: (1) an *error-monitor* to find out which items (e.g., submission date, price of real estate, etc.) are wrong versions according to the reports from the crowds, which belongs to a *passive crowdsourcing* strategy; (2) a *task-manager* to allocate the tasks to human workers intelligently; (3) a *smart-decision-maker* to identify which answer from the crowds is correct with *active crowdsourcing* methods; and (4) a *whom-to-ask-finder* to discover which users (or human workers) should be the most credible according to their answer records.

I. Introduction

In the big data era, we are experiencing a boosting of Web services, and the data on the Web is playing an important role. Moreover, among all such services, the data from one type of them is observed with frequent updates, which is termed as *multi-version data on the Web* [1]. For example, the latest values of real estate, deadlines of call-for-papers, and current fuel prices in different districts, etc.

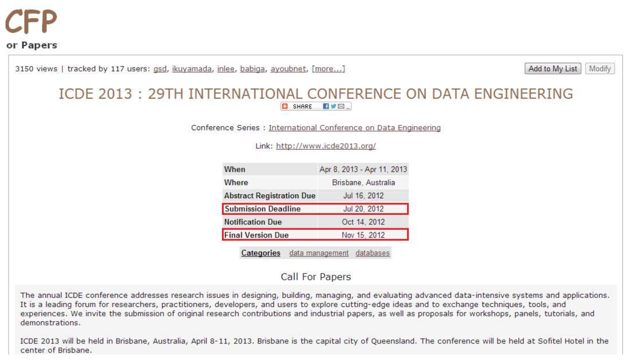
In order to maintain these latest updates, some automatical Web data integration systems have been developed. Most of them adopt the two-phase paradigm: web crawling-based automated information extraction and data integration. Although many machine-learning-based techniques are incorporated into these systems to enhance data quality [2], the incorrect and inconsistent multi-version data usually appears due to the delays in updates and errors while integrating updates. In the following, we use two real examples to illustrate the errors in multi-version data on the Web.

Example 1: (Deadlines of Call-For-Papers) Call-for-papers data is a representative multi-version data on the Web due to the frequent updates of the corresponding deadlines. Figure 1(a) is the screenshot of the current call-for-paper information of the ICDE 2013 conference in *WikiCFP* website, which is a best-known Web data integration system about data of call-for-papers. It includes the conference name, a link of its official website, important dates, conference location, and more detailed description. We can find that the submission deadline, which is highlighted by a red rectangle, is July 20th, 2012. However, the actual deadline is July 23rd according to the ICDE 2013 official website. Similarly, the final version due date is also incorrect.

Example 2: (Latest Retail Grocery Prices) The grocery prices fluctuates on a daily basis due to transportation, storage and many other reasons; and the prices from different retailers vary according to different business situations and policies. Figure 1(b) is the screenshot of the price information about one product of Celestial Tea, taken from a realtime integrator of Web grocery information named *mysupermarket.com*. As highlighted in Figure 1(b), the product price decreases greatly today(Sep 6th) and a change of price from *Walgreens* is just reported. However, the website tracks only three major retailers, and the price frequently observes delays. On the contrary, certain domestic retailers may provide more competitive price, but the integration of such physically distributed information entails more active surveillance.

In the aforementioned examples, these errors arise from data integration errors, update delays, or both. Thus, it is very difficult for the existing data cleaning approaches [3], [4] to guarantee the high quality muti-version data by only relying on machines. For example, it is hard to obtain the editing rules and master data [4] because of the uncertainty of updates. In reality, most existing techniques, like the integrity constraints-based methods and the dependence-based methods, assume that data sources are static and have no potential changes. Thus, they cannot handle the update delay errors.

To improve the quality of multi-version data in Web data integration system, we propose a novel data cleaning platform, called *CrowdCleaner*, via crowdsourcing approaches, which has been considered as a promising solution for the data cleaning and integration problems [5]. In other words, based on the designed platform, we do not only organize human workers to detect and repair false or delay versions of updates but also automatically determine which version of data should be accepted. Different from most of the recent work of crowdsoucing-based data management, which requires the users to specify a query, such as ranking [6], join [7], etc., to crowds (*a.k.a. human workers*), both users and human workers are likely the same since users find errors (which means that users give queries) and repair the errors (which represents that human workers finish jobs). Therefore, the most existing work may be called *active crowdsourcing* because users actively send queries to crowds, but our platform uses an *active-and-passive* hybrid crowdsourcing framework. To implement the hybrid framework, we face the following two challenges.


CFP

or Papers

3150 views | tracked by 117 users: gsd, ikuyamada, inlee, babiga, ayoubnet, [more...]

Add to My List | Modify

ICDE 2013 : 29TH INTERNATIONAL CONFERENCE ON DATA ENGINEERING

Conference Series : International Conference on Data Engineering

Link: http://www.icde2013.org/

When	Apr 8, 2013 - Apr 11, 2013
Where	Brisbane, Australia
Abstract Registration Due	Jul 16, 2012
Submission Deadline	Jul 20, 2012
Notification Due	Oct 14, 2012
Final Version Due	Nov 15, 2012

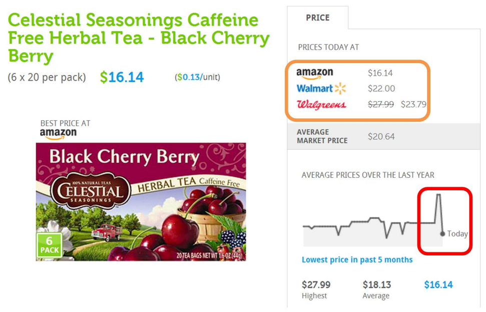
Categories data management databases

Call For Papers

The annual ICDE conference addresses research issues in designing, building, managing, and evaluating advanced data-intensive systems and applications. It is a leading forum for researchers, practitioners, developers, and users to explore cutting-edge ideas and to exchange techniques, tools, and experiences. We invite the submission of original research contributions and industrial papers, as well as proposals for workshops, panels, tutorials, and demonstrations.

ICDE 2013 will be held in Brisbane, Australia, April 8-11, 2013. Brisbane is the capital city of Queensland. The conference will be held at Sofitel Hotel in the center of Brisbane.

(a) Call-For-Paper of ICDE 2013 in the WikiCFP website


Celestial Seasonings Caffeine Free Herbal Tea - Black Cherry Berry

(6 x 20 per pack) $16.14 ($0.13/unit)

BEST PRICE AT amazon

PRICE

PRICES TODAY AT

amazon	$16.14
Walmart	$22.00
Walgreens	~~$27.99~~ $23.79
AVERAGE MARKET PRICE	$20.64

AVERAGE PRICES OVER THE LAST YEAR

Lowest price in past 5 months

$27.99 Highest | $18.13 Average | $16.14

(b) Update Delay of the Price of Celestial Tea

Fig. 1. Errors in Multi-version Data on The Web

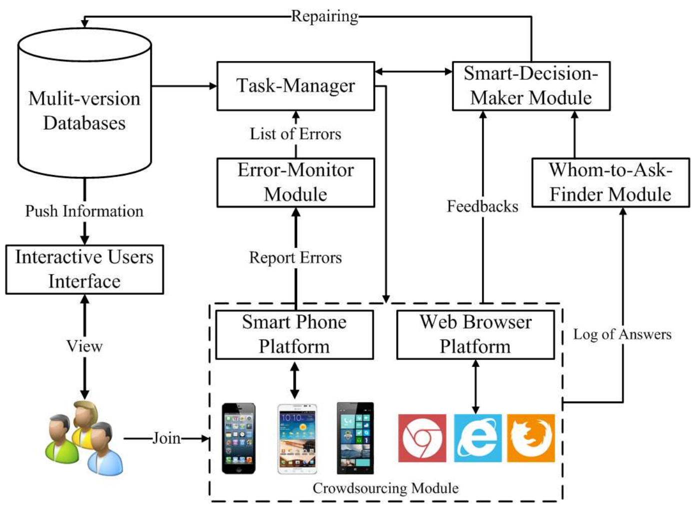


Fig. 2. The CrowdCleaner Architecture

Challenge 1: Which repaired version of data is correct when human workers submit several different suggestions? In *CrowdCleaner* platform, for an incorrect multi-version data (*e.g., submission date in a call-for-paper*), human workers may submit several different suggestions for repairing. Thus, the repaired suggestion may be considered as a discrete random variable, and then we use *entropy* to measure the consistency of the suggestions. If these suggestions are inconsistent, this system will consult with related experts. Otherwise, the system will repair the item according to the suggestion associated with the highest probability.

Challenge 2: Which human workers are the most credible? To save the consultation cost, could we ask the most credible human workers instead of consulting with experts? In other words, for incorrect information, which human workers should be prioritized to be asked by the system? As more and more users participate in the system, we collect many logs of human workers and propose a probabilistic model to analyze which group of human workers is the most credible. More details will be introduced in Section 2.

In the following sections, we first introduce *CrowdCleaner*, and then outline what functionalities we shall demonstrate.

II. CrowdCleaner Prototype

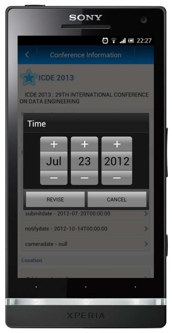
In this section, we propose the novel data cleaning platform, called *CrowdCleaner*, for multi-version data on the Web. The system architecture of *CrowdCleaner* is introduced in Section II-A. Furthermore, the critical techniques of *CrowdCleaner* are described in Section II-B.

A. System Architecture

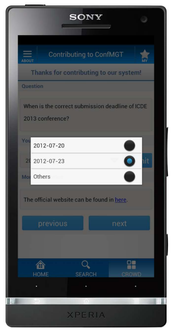
The architecture of *CrowdCleaner* is depicted in Figure 2. *CrowdCleaner* adopts the crowdsourcing strategy to inspect and repair the incorrect items. In contrast to the most existing *active* crowdsouring methods, which require that users actively provide questions or queries to human workers, *CrowdCleaner* system uses an *active-and-passive* hybrid crowdsourcing framework. On the one hand, the *error-monitor* module passively waits for the error reports from human workers rather than the system actively identifies errors. The passive approach can minimize the monitoring cost due to too much delay and incorrect versions of updates for multi-version data. On the other hand, after the errors are collected, the *task-manager* module actively assigns these error-repair tasks to the crowds. According to the feedbacks from human workers, the *smart-decision-maker* module evaluates which suggestion for repairing is correct. Meanwhile, the *whom-to-ask-finder* module also analyzes which workers are the most credible.

As a crowdsourcing-based system, how to encourage crowds to finish tasks is always the primary problem. *CrowdCleaner* designs a credits-based scheme to motivate human workers. In this scheme, each user is assigned with some credits when his or her system account initializes. Then, users have to spend their credits when he or she tries to report an error, but these spent credits and some bonus credits are returned by the system if his or her report is correct. Moreover, a user can also earn credits if he or she performs a crowdsorucing task correctly. In *CrowdCleaner*, users are encouraged to accumulate credits to build up their prestige. According to the credits-based scheme, we hope to avoid the spam answers as many as possible, meanwhile, the credible human workers can be motivated. In the following paragraphs, we briefly present the four core components in *CrowdCleaner*.

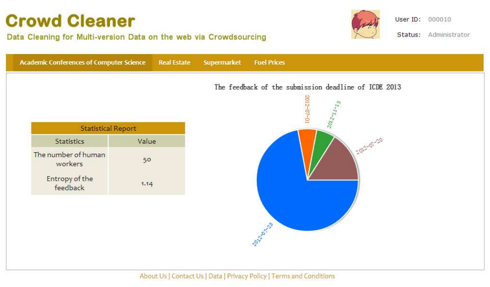
Error-monitor. It discovers the new errors of multi-version data and evaluates whether each reported error is valuable. When human workers report the errors, the error-monitor module is woken up. Then, the *error-monitor* module determines which reported errors are the actual errors, or the spam reports, according to the function which involves the users' credits, frequency of the reported errors, and several other parameters. Finally, only the errors passing the evaluation of the *error-monitor* module are submitted to the *task-manager* module.


SONY
22:27
Conference Information
ICDE 2013
ICDE 2013 : 29TH INTERNATIONAL CONFERENCE ON DATA ENGINEERING
Time
Jul
23
2012
REVISE
CANCEL
submitdate - 2012-07-20T00:00:00
notifydate - 2012-10-14T00:00:00
cameradate - null
Location
XPERIA

(a) An Error Report

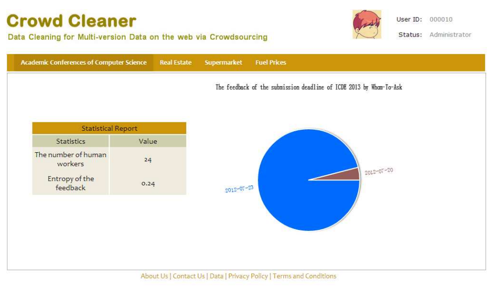

SONY
ABOUT
Contributing to ConfMGT
MY
Thanks for contributing to our system!
Question
When is the correct submission deadline of ICDE 2013 conference?
2012-07-20
2012-07-23
Others
The official website can be found in here.
previous
next
HOME
SEARCH
CROWD
XPERIA

(b) A Task for Workers


Crowd Cleaner
Data Cleaning for Multi-version Data on the web via Crowdsourcing
User ID: 000010
Status: Administrator
Academic Conferences of Computer Science
Real Estate
Supermarket
Fuel Prices
The feedback of the submission deadline of ICDE 2013
2012-07-01
2012-11-13
2012-07-20
2012-07-23
About Us | Contact Us | Data | Privacy Policy | Terms and Conditions

Statistical Report	
Statistics	Value
The number of human workers	50
Entropy of the feedback	1.14

(c) The Feedback from General Crowds


Crowd Cleaner
Data Cleaning for Multi-version Data on the web via Crowdsourcing
User ID: 000010
Status: Administrator
Academic Conferences of Computer Science
Real Estate
Supermarket
Fuel Prices
The feedback of the submission deadline of ICDE 2013 by Whom-To-Ask
2012-07-20
2012-07-23
About Us | Contact Us | Data | Privacy Policy | Terms and Conditions

Statistical Report	
Statistics	Value
The number of human workers	24
Entropy of the feedback	0.24

(d) The Feedback from a Credible Crowd

Fig. 3. Functionalities of *CrowdCleaner*

Task-manager. It assigns the questions to human workers based on the submitted errors from the *error-monitor* module. The *task-manager* module first posts the corresponding questions to the crowds. If the feedbacks from the crowds have a big discrepancy, the *task-manager* receives the requirement from the *smart-decision-maker* module and sends the question to the experts according to the constraint conditions (e.g., consultation cost budget). In addition, after the *whom-to-ask-finder* module discovers some credible human workers, the *task-manager* prefers to allocate tasks to these credible workers, so that the risk of spam workers can be reduced. Besides the task assignment, the *task-manager* module assists the *whom-to-ask-finder* module to train human workers' creditability. Sometimes the *task-manager* module allocates some tasks with known ground truths to train the credible human workers for the *whom-to-ask-finder* module.

Smart-decision-maker. This is the most important module of *CrowdCleaner*. It not only interacts with the *task-manager* and *whom-to-ask-finder* modules but also determines the final repaired results. Even though the credits-based scheme can reduce spam answers, the feedbacks from different human workers can be still inconsistent. Thus, each expected repaired result is actually considered as a discrete random variable. The *smart-decision-maker* module employs the *entropy-based decision strategy*, whose details will be introduced in Section II-B, to determine whether the answers of human workers are consistent. If repaired answers are inconsistent, the *smart-decision-maker* module requests the *task-manager* module to consult with the experts in order to ensure which answer is correct. Otherwise, the feedback with the most votes is the repaired result. Moreover, *CrowdCleaner* maintains a list of experts, who are some computer scientists and our system administrators. Thus, the feedbacks from the experts are always correct because these repaired answers are manually repaired by the experts. In addition, *CrowdCleaner* ignores the conflicts of experts' feedbacks since the latest version of a multi-version data is usually unique.

Whom-to-ask-finder. This module is another important component and contribution of *CrowdCleaner*. As mentioned in Section I, the multi-version data on the Web easily generates update delay errors due to frequent updates. Thus, it is usually impractical to wait for a long period (e.g. one week) in order to fix the errors. However, the experts are often busy and may not be able to provide feedback immediately. In other words, it is not always efficient to consult with the experts. In order to solve this challenge, *CrowdCleaner* designs the *who-to-ask-finder* module, which finds some credible human workers instead of experts. Based on the credits of human workers, this module employs a majority-voting-based model to determine which group of human workers has the highest confidence under the given constraints (e.g., the maximum inquiry number of human worker according to the average responding time)[8], [9]. It makes sense to set the constraints so that the repair is valuable, otherwise the more updates would happen before our repair. More detailed techniques regarding the *whom-to-ask-finder* module are discussed in Section II-B.

B. Critical Techniques

In this subsection, we mainly introduce two critical techniques, *entropy-based decision strategy* and *whom-to-ask-finder*, in the *CrowdCleaner* system.

Entropy-based decision strategy. As discussed above, for an error, the repaired suggestions from different human workers maybe inconsistent. According the frequencies of different suggestions, the possibility of each suggestion, x_i ($1 \leq i \leq n$), is denoted $Pr(x_i)$. Formally, we define the entropy of an expected repaired result, X, as follows.

$$H(X) = -\sum_{i=1}^{n} Pr(x_i) log Pr(x_i) \tag{1}$$

We adopt the entropy to measure the diversity of repaired suggestions from crowds. When the diversity is too large, we further use the submodularity of entropy to clean the uncertainty of spam suggestions. More details of *entropy-based decision strategy* can be found in [10].

Whom-to-ask-finder. The *who-to-ask-finder* module is to find some credible human workers instead of experts. Formally, a group of credible workers is a set of workers $CW_n = \{cw_1, cw_2, \ldots, cw_n\} \subseteq W$ with size n, where each cw_i is associated with an individual confidence c_i, and W is the set of all human workers. Then, we can define the *group confidence of credible workers* as follows.

$$\begin{aligned} GC(CW_n) &= \Pr(|C| \geq \lceil \frac{n}{2} \rceil) = \Pr(|C| \geq \frac{n+1}{2}) \\ &= \sum_{k=\lceil \frac{n}{2} \rceil}^{n} \sum_{A \in F_k} \prod_{i \in A} c_i \prod_{j \in A^c} (1-c_j) \end{aligned}$$

where $|C|$ the number of human workers who give the correct suggestion, and $F_k = \{A||A| = k, A \subseteq CW_n\}$ is all subsets of CW_n with size k and A^c is the complementary set of A.

The *group confidence of credible workers* is used to measure which human workers are credible. Furthermore, we design a divide-and-conquer algorithm and an effective bounding technique to optimize the computation of group confidence of credible workers. More detailed can be found in [8], [9].

III. Demonstration Overview

In this section, we describe a variety of scenarios of *CrowdCleaner* platform in detail, and explain the aims of our demonstration. The CrowdCleaner system is enabled to tackle a wide range of multi-version data cleaning challenges, including deadlines in call-for-papers, latest prices of goods in supermarkets, current prices of real estate and so on. However, for brevity's sake, we illustrate the functionalities of the system based on the process of cleaning call-for-papers data.

As discussed in Example 1, the submission deadline of ICDE 2013 is incorrect. We specially keep this error in *CrowdCleaner* system for this demo. In particular, based on the error of the submission deadline of the ICDE 2013, we exhibit the following four scenarios of *CrowdCleaner* to the audiences. (1) how *CrowdCleaner* monitors the error reports from crowds with the aid of the user-friendly interface (Figure 3(a)); (2) how human workers work on the crowdsourcing platform? In other words, how the *task-manager* module allocates questions to the human workers effectively (Figure 3(b)); (3) how *CrowdCleaner* identifies which feedback should be the correct repaired result (Figure 3(c)); (4) how the credible human workers discovered by the *whom-to-ask-finder* can help *CrowdCleaner* enhance the quality of repaired results.

Error-monitor. We shall demonstrate how *CrowdCleaner* obtains the error reports from users. Continued on Example 1, Figure 3(a) shows that a user is reporting the error of the submission deadline of the ICDE 2013 and providing a correct suggestion, July 23rd 2012, for repairing this error. After sending this error report, the *error-monitor* module is woken up. We assume that the user has high credits, so the *error-monitor* module will activate the *task-manager* module to execute the crowdsourcing-based operation.

Task-manager. After receiving the reported errors, the *task-manger* module allocates the corresponding problems to the crowds. In Figure 3(b), *task-manger* sends a question, "when is the correct submission deadline of the ICDE 2013?", to a worker by the smart-phone platform. The worker can choose whether to answer this question. If he chooses to answer, the worker needs to select or input a correct date. Otherwise, he can answer other questions via touching the "next" button. In particular, *CrowdCleaner* also provides the link of the official website for answering the question as easily as possible.

Smart-decision-maker. After questions are assigned to human workers by the *task-manager* module, the *smart-decision-maker* module collects the feedbacks from workers. For example, Figure 3(c) is the screenshot of the system administrator to show the distribution of feedbacks from 50 random human workers. Although more than half of the workers select the correct date, July 23rd, other workers provide different feedbacks. The entropy of this expected repaired answer is 1.14. If the threshold is 0.6, *CrowdCleaner* consults with experts.

Whom-to-ask-finder. In order to avoid the feedback delay from the experts, the *whom-to-ask-finder* module can discover the credible workers as the substitutes of experts. For the same question, "when is the correct submission deadline of the ICDE 2013?", if the *task-manager* pushes the question to credible workers, the distribution of feedbacks from 24 credible workers is shown in Figure 3(d). We observe that the majority of 24 workers select July 23rd as the correct answer. The entropy of the expected repaired answer from these credible workers is very low. Therefore, *whom-to-ask-finder* can reduce the risk of spam human workers and help *CrowdCleaner* enhance the data quality.

IV. Conclusions

In this demo, we present a novel data cleaning prototype platform, called *CrowdCleaner*, for cleaning multi-version data on the web via crowdsourcing. Since multi-version data easily leads to the update delay and the data integration errors, most existing data integration and cleaning approaches do not work. In order to wipe out the errors of multi-version data, *CrowdCleaner* adopts an (*active-and-passive*) hybrid crowdsourcing framework, which consists of the following four modules. (1) *Error-monitor*. In contrast to the recent *active* crowdsourcing methods, the error-monitor module uses the *passive* crowdsourcing method to wait for the human workers to report errors so that the monitoring cost is minimized. (2) *Task-manager*. It allocates the tasks to the crowds intelligently and assists *whom-to-ask-finder* to train the human workers' creditabilities. (3) *Smart-decision-maker*. It aims to identify which suggestion for repairing is correct. The *entropy-based decision strategy* is proposed to measure the consistency of the suggestions from the crowds. If the suggestions are inconsistent, it consults with the experts, otherwise the highest probability suggestion is selected as the repaired result. (4) *Whom-to-ask-finder*. In order to save the consulting cost, it can obtain the credibility information of human workers to replace experts according to the answers log of the crowds.

V. Acknowledgments

This work is supported in part by the Hong Kong RGC Project MHKUST602/12, National Grand Fundamental Research 973 Program of China under Grant 2012-CB316200, Microsoft Research Asia Gift Grant and Google Faculty Award 2013.

References

[1] W. Le, F. Li, Y. Tao, and R. Christensen, "Optimal splitters for temporal and multi-version databases," in *SIGMOD*, 2013.
[2] L. Jiang, Z. Wu, Q. Feng, J. Liu, and Q. Zheng, "Efficient deep web crawling using reinforcement learning," in *PAKDD*, 2010.
[3] X. L. Dong, L. Berti-Equille, and D. Srivastava, "Integrating conflicting data: The role of source dependence," *PVLDB*, 2009.
[4] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu, "Towards certain fixes with editing rules and master data," *PVLDB*, 2010.
[5] A. Doan, A. Y. Halevy, and Z. G. Ives, *Principles of Data Integration*. Morgan Kaufmann, 2012.
[6] S. Guo, A. G. Parameswaran, and H. Garcia-Molina, "So who won?: dynamic max discovery with the crowd," in *SIGMOD*, 2012.
[7] A. Marcus, E. Wu, D. R. Karger, S. Madden, and R. C. Miller, "Human-powered sorts and joins," *PVLDB*, 2011.
[8] C. C. Cao, J. She, Y. Tong, and L. Chen, "Whom to ask? jury selection for decision making tasks on micro-blog services," *PVLDB*, 2012.
[9] C. C. Cao, Y. Tong, L. Chen, and H. V. Jagadish, "Wisemarket: a new paradigm for managing wisdom of online social users," in *KDD*, 2013.
[10] C. J. Zhang, L. Chen, H. V. Jagadish, and C. C. Cao, "Reducing uncertainty of schema matching via crowdsourcing," *PVLDB*, 2013.