Crowdsourcing Entity Resolution: a Short Overview and Open Issues

Xiao Chen
Otto-von-Guericke University
Magdeburg
xiao.chen@ovgu.de

ABSTRACT

Entity resolution (ER) is a process to identify records that stand for the same real-world entity. Although automatic algorithms aiming at solving this problem have been developed for many years, their accuracy remains far from perfect. Crowdsourcing is a technology currently investigated, which leverages the crowd to solicit contributions to complete certain tasks via crowdsourced marketplaces. One of its advantages is to inject human reasoning to problems that are still hard to process for computers, which makes it suitable for ER and provides an opportunity to achieve a higher accuracy. As crowdsourcing ER is still a relatively new area in data processing, this paper provides an overview and a brief classification of current research state in crowdsourcing ER. Besides, some open issues are revealed that will be a starting point for our future research.

General Terms

Theory

Keywords

Crowdsourcing, Entity Resolution, Record Linkage

1. INTRODUCTION

Entity resolution (ER) is a process to identify records that refer to the same real-world entity. It plays a vital role not only in traditional scenarios of data cleaning and data integration, but also in web search, online product comparisons, etc. Various automatic algorithms have been developed in order to solve ER problems, which generally include two classes of techniques: similarity-based and learning-based approaches. Similarity-based techniques use similarity functions, where values of record pairs similarity is above a preset threshold are considered to be matched. Learning-based techniques use machine learning modeling ER as a classification problem and are training classifiers to identify matching and non-matching record pairs [14]. However, the accuracy of both classes of computer-based algorithms is still far from perfect, particularly for big data without fixed types and structures.

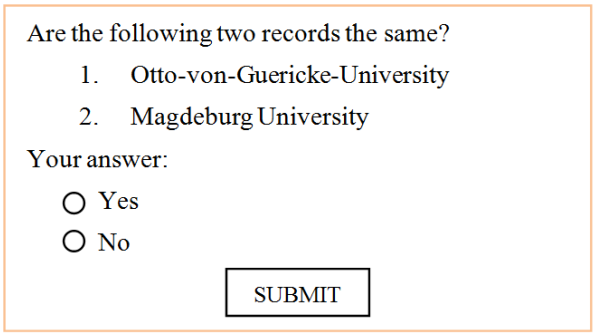

Figure 1: A HIT for ER

Crowdsourcing was first introduced in the year of 2006 [6] and is gaining growing interest in recent years. Current approaches investigate how Crowdsourcing is suitable to improve the accuracy of ER, because people are better at solving this problem than computers. Although the research on crowdsourcing ER was only started in recent years, there have been several significant research contributions. This paper gives an overview on the current research state of crowdsourcing ER, classifies and compares the related research, and tries to point out some open issues.

The rest of the paper is organized as follows. As crowdsourcing is still not fully established as a technology in the database community, in Section 2 background information on crowdsourcing ER are presented. Then in Section 3, the current state of the research in crowdsourcing ER is presented, different research approaches are classified and compared. After that, open issues are presented in Section 4. Finally, a conclusion is presented in Section 5.

2. CROWDSOURCING

In their early days, computers were mainly used for specific domains that required strong calculation powers. But up until today many tasks, especially those requiring complex knowledge and abstract reasoning capabilities, are not well supported by computers. Crowdsourcing derives its name from "crowd" and "sourcing", as it outsources tasks to the crowd via the Internet. Many crowdsourced marketplaces, which are represented for instance by Amazon's Me-

27^{th} GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), 26.05.2015 - 29.05.2015, Magdeburg, Germany.

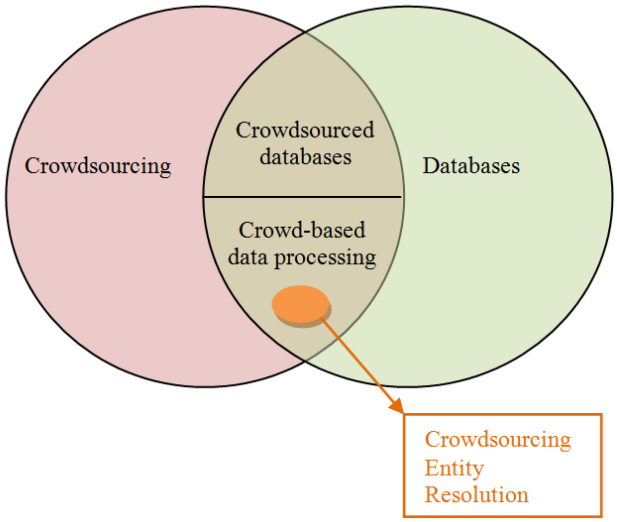

Figure 2: Research directions between crowdsourcing and databases

chanical Turk (MTurk), provide convenience for companies, institutions or individuals recruiting large numbers of people to complete tasks that are difficult for computers or cannot performed well by computer with acceptable effort. Entity resolution is one of such tasks, which can apply crowdsourcing successfully. People are better at solving ER than computers due to their common sense and domain knowledge. For example, for "Otto-von-Guericke-University" and "Magdeburg University" it is easy for people around Magdeburg to know that both of them refer to the same university, while for computer algorithms they are very likely to be judged as different universities. The tasks on MTurk are called Human Intelligence Tasks (HITs). Figure 1 depicts a possible HIT example of ER. One HIT is usually assigned several workers to guarantee the result quality. Small amounts of money, e.g. 3-5 cents are typical at MTurk, are paid for each worker per HIT.

The current research on crowdsourcing and databases covers two aspects (see Figure 2): on the one hand, databases should be adjusted to support crowd-based data processing (see [3, 10, 11]); on the other hand, crowd-based technology can help to make more effective and broader data processing possible (see [2, 5, 8, 9]). More effective data processing means that the returned query results could be more accurate after leveraging crowdsourcing. Besides, traditional databases cannot handle certain queries such as incomplete data queries or subjective operation queries. By using crowdsourcing, the scope of queries to be answered is broadened. Crowdsourcing ER belongs to the second area, i.e., crowd-based data processing. Specifically, crowdsourcing ER is an important part of join queries. Join queries allow establishing connections among data contained in different tables and comparing the values contained in them [1]. This comparison is not necessarily simple, since there are different expressions for the same real-world fact, more cases with the comparisons among different media and more cases that need human's subjective comparison. Then ER is a necessary step to better answer the join queries.

3. OVERVIEW AND CLASSIFICATION OF CURRENT RESEARCH STATE

Figure 3 presents a classification for the main research on crowdsourcing ER. From the perspective of crowdsourcing, most of the researches uses crowdsourcing only for identifying matching record pairs. Only one recent approach proposes leveraging crowdsourcing for the whole process of ER, which gives a novel and valuable view on crowdsourcing ER. The workflows of approaches, which leverage crowdsourcing solely for identifying matching record pairs, vary widely. In general, the workflow of crowdsourcing ER has been developed step by step. From the beginning, ER is proposed to be solved by crowdsourcing only, until now a much more complete workflow is formed by integrating different research (see Figure 4). In addition, these approaches focus on different problems. Many novel ideas and algorithms have been developed to optimize crowdsourcing ER. Gokhale et al. presented the Corleone approach and tried to leverage crowdsourcing for the whole process of ER, which is described as hands-off crowdsourcing [4].

In Subsection 3.1 the research leveraging crowdsourcing only for the identification of matching record pairs is described. The corleone approach, which leverages crowdsourcing for the whole process of ER, is then discussed in Subsection 3.2.

3.1 Crowdsourcing for Identifying Matching Record Pairs

Figure 4 depicts a complete hybrid workflow for ER, which contains all proposals to optimize the workflow in crowdsourcing ER. Instead of letting crowds answer HITs that contain matching questions for all records, the first step in the complete workflow is to choose a proper machine-based method to generate a candidate set that contains all record pairs with their corresponding similarities. Then pruning is performed to reduce the total number of required HITs. In order to further reduce the number of required HITs, the transitive relation is applied in the procedure of crowdsourcing, i.e., if a pair can be deduced by transitive relation, it does not need to be crowdsourced. For instance, given three records a, b, and c, the first type of transitive relation is, if a matches b and b matches c, then a matches c. The other type of transitive relation means, if a matches b and b does not match c, then a does not match c. After all record pairs are further identified by crowdsourcing or transitive relations, a global analysis can be performed on the initial result. The global analysis was suggested by Whang et al. [16]. They apply transitive relations only as an example of a global analysis after the process of crowdsourcing. In the case of the integrated ER workflow, which applies transitive relations during the process of crowdsourcing, its global analysis may be implemented by other techniques such as correlation clustering to further improve the result [16]. After a global analysis, the final result is obtained.

The three segments in the dashed box are optional, i.e., in some research, some of them are not included. In the following, specific workflows for different approaches are presented and their contributions are summarized. One important assertion needs to be pointed out: most of the research is done given the assumption that people do not make mistakes when answering HITs. Therefore, each HIT is only assigned to one worker. The problems caused by mistakes

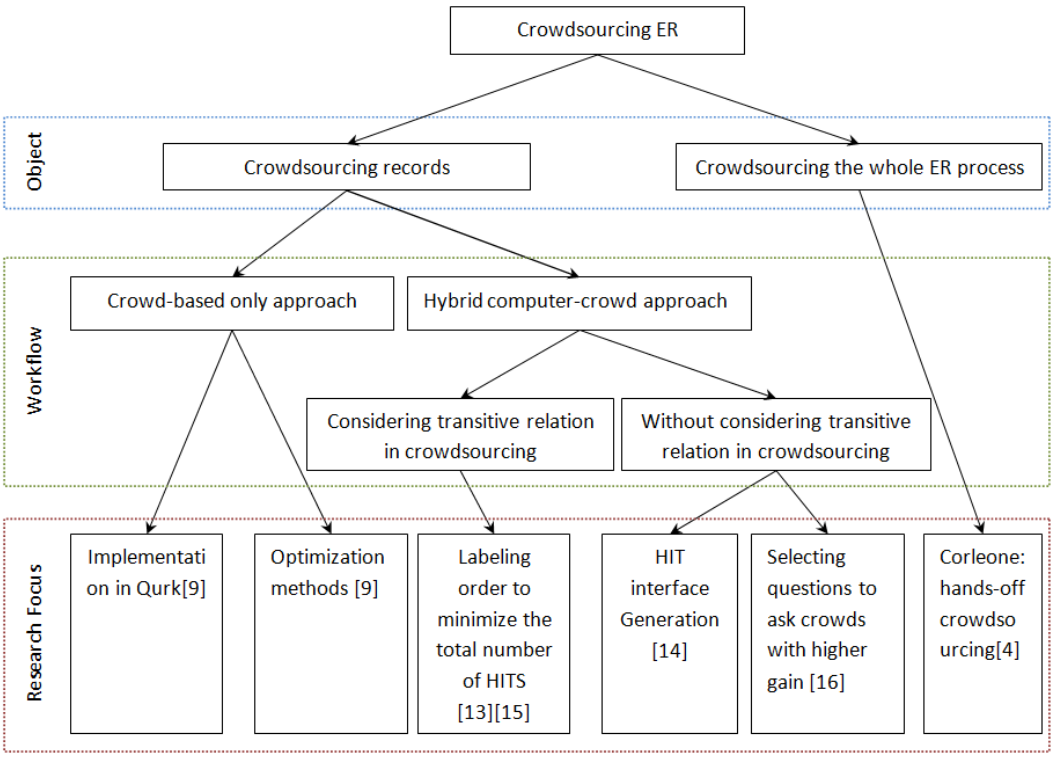

Figure 3: Classification for the research on crowdsourcing ER

that crowds may make are addressed for instance in [7, 12].

3.1.1 *Crowd-based Only ER*

Marcus et al. proposed to solve ER tasks only based on crowdsourcing [9]. They suggested to using Qurk [10], a declarative query processing system to implement crowd-based joins, and the workflow for ER is only crowd-based, i.e., crowdsourcing is used to complete the whole process of ER. Correspondingly, their workflow does not contain any segment of the three dashed boxes. As mentioned in Section 2, although crowdsourcing can improve the accuracy of ER, it causes monetary costs and is much slower than automatic algorithms. In order to save money and reach lower latency, Marcus et al. proposed two optimization methods, summarized here.

Batching:.

the basic interface of crowdsourcing ER is similar to Figure 1, which asks workers to answer only one question in each HIT and is called simple join. The question in simple join is pair-based, i.e., ask workers whether two records belong to the same entity. For a task to identify the same entities from two sets of records respectively with m and n records, m*n HITs are needed. Two optimizations are provided. One is called naive batching. It asks workers to answer b questions in each HIT. Each question in it is also pair-based. In this way, the total amount of HITs is reduced to (m*n)/b. The other one is called smart batching. Instead of asking workers whether two records belong to the same entity or not, it asks workers to find all matching pairs from two record lists. If the first list contains a records that are selected from one data set and the second list contains b records that are selected from the other data set, the total number of HITs are (m*n)/(a*b).

Feature filtering optimization:.

for some kinds of records, certain features are useful for being join predicates. For instance, there are two groups of people photos, and if each photo is labeled with the gender of the person on it, only photos with a matching gender are necessary to be asked. The feature should be extracted properly, or it leads to large amounts of unnecessary label work and is unhelpful to reduce the total number of HITs.

Even though the above described optimization methods are applied for ER, the crowd-based only workflow cannot satisfy the development for larger and larger data sets. Therefore, the research in rest of this subsection tends to develop a hybrid workflow for crowdsourcing ER.

3.1.2 *Hybrid Computer-Crowd Entity Resolution Without Considering Transitive Relation During the Process of Crowdsourcing*

Wang et al. proposed an initial hybrid workflow [14], which contains only the segment in the red dashed box. The pruning technique in it is simply abandoning the pairs with similarities under a threshold. It describes two types of HIT generation approaches: pair-based HIT generation and cluster-based HIT generation. These two types are actually similar to naive batching and smart batching mentioned above in the optimization part of [9]. A pair-based HIT consists of k pairs of records in a HIT. k is the permitted number of record pairs in one HIT, because too many

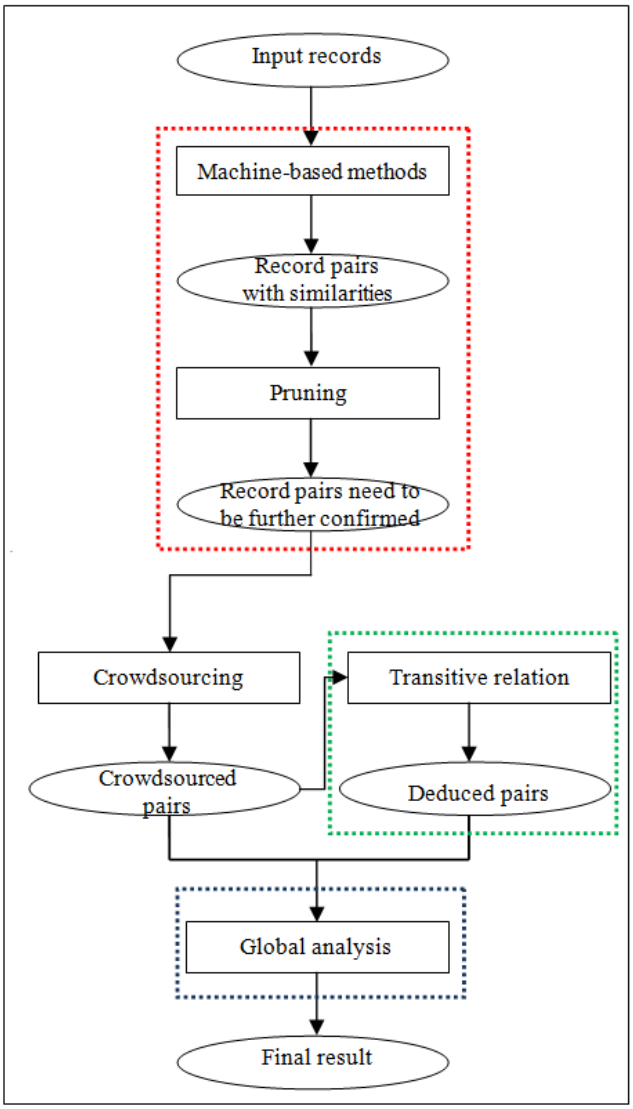

Figure 4: Complete workflow of crowdsourcing ER

pairs may lead to accuracy reductions of crowds‘ answers.

A cluster-based HIT consists of a group of k individual records rather than pairs but based on the given pairs. Wang et al. defined the cluster-based HIT generation formally and proves that this problem is NP-hard. Therefore, they reduced the cluster-based HIT generation problem to the k-clique edge covering problem and then apply an approximation algorithm to it. However, this algorithm fails on generating a minimum number of HITs. Therefore, a heuristic two-tiered approach is proposed to generate as few as possible cluster-based HITs. In addition, the following conclusion is made according to their own experimental results.

1. The two-tiered approach generates fewer cluster-based HITs than existing algorithms.
2. The initial hybrid workflow achieves better accuracy than machined based method and generates less HITs than the crowd-based only workflow.
3. The cluster-based HITs provide less latency than pair-based. HITs.

Wang et al. proved that cluster-based HITs have lower latency than pair-based HITs under the premise of reaching the same accuracy as pair-based HITs, which provides a baseline to prefer cluster-based methods for HIT design. However, the introduced approximation algorithm performs worse than a random algorithm, so it is not presented in this paper.

Whang et al. added a global analysis step to the initial workflow introduced above by Wang et al., but they did not consider using transitive relations to reduce the number of HITs either [16]. The global analysis after the process of crowdsourcing uses transitive relations as an example and permits other techniques, such as correlation clustering to improving the accuracy. Whang et al. focused their research on developing algorithms to ask crowds HITs with the biggest expected gain. Instead of simply abandoning the pairs with similarity scores under a threshold, which is adopted by Wang et al. [14], it develops an exhaustive algorithm, which computes the expected gain for asking crowds questions about each record pair and then chooses the question with the highest estimated gain for crowdsourcing. The exhaustive estimation algorithm is #P-hard. Therefore, the proposed GCER algorithm produces an approximate result within polynomial time and includes the following optimization on the exhaustive algorithm:

1. It only computes the expected gain of record pairs with very high or low similarities.
2. It uses the Monte-Carlo approximation to substitute the method of computing the expected gain for one record pair.
3. The results of the calculations that have already been made before can be shared to the later calculation, instead of recalculation.
4. Instead of resolving all records, only resolving records that may be influenced by the current question.

At last, because the GCER algorithm is still complex, it proposes a very simple half algorithm to choose questions for crowdsourcing. The half algorithm chooses the record pairs with a matching probability closest to 0.5 to ask crowds whether they match or not. In summary, it is non-trivial that Whang et al. uses transitive relations to remedy the accuracy loss caused by such HITs that people cannot answer correctly. However, Whang et al. have not further applied transitive relations during the process of crowdsourcing. Besides, although several optimizations are proposed to improve the performance of the exhaustive algorithm, the complexity of the algorithm is still very high and infeasible in practice.

3.1.3 *Hybrid Computer-Crowd ER Considering Transitive Relation During the Process of Crowdsourcing*

Vesdapunt et al. [13] and Wang et al. [15] considered the transitive relation in the process of crowdsourcing. Therefore, their workflows contain the segments in the red and green dashed boxes in Figure 4. Their research defines the problem of using transitive relations in crowdsourcing formally and proposes a hybrid transitive-relations and crowdsourcing labeling framework. The basic idea is that, if a

record pair can be deduced according to the matching results that are obtained by crowdsourcing, there is no need to crowdsource it. A record pair can be deduced if transitive relations exist. Two types of transitive relations are defined: positive transitive relations and negative transitive relations, which refer to the two types of transitive relations described at the first paragraph of Section 3.1. A record pair x_1, x_n can be deduced to be a matching pair, only if all pairs among the path from x_1 to x_n are matching pairs. A record pair x_1, x_n can be deduced to be a non-matching pair, only if there exist one non-matching record pairs among the path from x_1 to x_n.

The number of HITs is significantly effected by the labeling order because of applying transitive relations for crowdsourcing ER, i.e. in their work Vesdapunt et al. suggest to first ask crowds questions for real matching record pairs and then for real non-matching record pairs. Because whether two records match or not in reality cannot be known, HITs are first generated for record pairs with higher matching probabilities. However, the latency that a question is answered by crowdsourcing is long, and it is not feasible to publish a single record pair for crowdsourcing and wait for its result to decide whether the next record pair can be deduced or has to be crowdsourced. In order to solve this problem, a parallel labeling algorithm is devised to reduce the labeling time. This algorithm identifies a set of pairs that can be crowdsourced in parallel and asks crowds questions about these record pairs simultaneously, then iterates the identification and crowdsourcing process according to the obtained answers until all record pairs are resolved. The proposed algorithm is evaluated both in simulation and a real crowdsourcing marketplace. The evaluation results show that its approaches with transitive relations can save more monetary costs and time than existing methods with little loss in the result quality.

However, Vesdapunt et al. proved that the algorithm proposed by Wang et al. may be $\Omega(n)$ worse than optimal, where n is the number of records in the database. This proof means the algorithm is not better than any other algorithms, because Vesdapunt et al. also prove that any algorithm is at most $\Omega(n)$ worse than optimal. Therefore, Vesdapunt et al. presented their own strategies to minimize the number of HITs considering transitive relations in the process of crowdsourcing. One simple strategy is to ask crowds questions of all record pairs in a random order without considering the matching probabilities of record pairs. Although this strategy is random, it is proved that it is at most $o(k)$ worse than the optimal, where k is the expected number of clusters. For both approaches a graph-clustering-based method is adopted to efficiently show the relations among possibly matching records. Since the expected number of clusters cannot exceed the number of records, the random algorithm is better than the algorithm proposed by Wang et al. Another strategy called Node Priority Querying is also proved to be at most $o(k)$ worse than the optimal. These algorithms are evaluated using several real-world data sets. The results in different data sets are not stable. However, overall the node priority algorithm precedes the random algorithm and the algorithm proposed by Wang et al. The random algorithm is superior to the algorithm proposed by Wang et al. in some cases.

In summary, both Vesdapunt et al. and Wang et al. considered transitive relations during the process of crowdsourcing, which opens a new perspective to further decrease the number of HITs to reduce the latency and lower the monetary cost. However, their proposed algorithms perform not stably on different data sets and can be optimized or redesigned.

3.2 Applying Crowdsourcing for the Whole ER Process

All approaches introduced so far apply crowdsourcing only for identifying whether record pairs are matching or not. The proposed algorithms have to be implemented by developers. Nowadays, the need for many enterprises to solve ER tasks is growing rapidly. If enterprises have to employ one developer for each ER task, for so many tasks the payment for developers are huge and not negligible. Even in some cases, some private users cannot employ developers to help them complete ER tasks by leveraging crowdsourcing, as they have only a small amount of money. In order to solve this problem, Gokhale et al. proposed hands-off crowdsourcing for ER, which means that the entire ER process is completed by crowdsourcing without any developer [4]. This hands-off crowdsourcing for ER is called Corleone, which can generate blocking rules, train a learning-based matcher, estimate the matching accuracy and even implement an iteration process by crowdsourcing. Each of the above mentioned aspects is settled into a module, and specific implementations are presented to each module.

4. OPEN ISSUES

In this section, open issues on crowdsourcing ER are presented that are from the authors perspective the most important ones to be addressed by future research.

Machine-based methods and pruning approaches: The machine-based methods used in most approaches are similarity-based and pruning methods are simply abandoning the record pairs, whose matching probabilities are above or below given thresholds. Such approaches cannot get satisfactory results in the case of more and more complex data environments, because record pairs with very high matching probabilities may refer to different entities, and vice versa. Therefore, machined-based methods should be extended to learning-based and pruning approaches should be rigorously designed to avoid abandoning the record pairs, which indeed need to be further confirmed.

Only transitive relations applied: currently, transitive relations have been widely adopted in crowdsourcing ER, which are not considered at the early stage of crowdsourcing ER. Other techniques such as correlation clustering could be applied to ER.

Limited comparability of results: both [13] and [15] develop strategies to minimize the number of HITs that are needed to be sent to crowdsourcing. Both of them evaluate their own algorithms using different data sets and compare the performance of their own algorithm with other existing algorithms. However, some of the evaluation results are inconsistent. The reason is that the same algorithm performs varies for different data sets perhaps leading to a different result. Therefore, research to study which algorithm is more suitable for

specific kinds of data sets is necessary for the development of crowdsourcing ER. In addition, new algorithms can be designed, which performs stably on different kinds of data sets.

Only initial optimization strategies: as leveraging crowdsourcing for the whole process of ER, i.e., hands-off crowdsourcing, is just in the beginning of its devopment. The following optimization techniques can be developed: first, the current hands-off crowdsourcing for ER is based on the setting of identifying record pairs from two relational tables, which may be extended to other ER scenarios. Second, the current strategy to extract samples for generating blocking rules is quite simple, and better sampling strategies should be explored.

Ontologies and indexes: once a decision is made, the knowledge injected by the crowd is widely lost. Another interesting possible research question could be, how this feedback could be gathered and described for instance by an ontology.

5. CONCLUSIONS AND FUTURE WORK

This paper gives an overview of the current research state in crowdsourcing ER. Most of the approaches focus on leveraging crowdsourcing to verify the matching of record pairs. From the early stage of the research to more recent approaches, the workflow was optimized step by step and more aspects were considered for the process of crowdsourcing, which developed from crowd-based only workflow to hybrid computer-crowdsourcing workflow, which considers transitive relations. However, this does not mean that the research on the initial workflow is less significant. In contrast, every approach is valuable and contributes to various research aspects of crowdsourcing ER.

Most recently, a novel perspective to crowdsourcing is proposed, which extends the crowdsourcing object and leverages crowdsourcing not only to verify the matching of record pairs, but also to implement the algorithms, train a learning-based matcher, estimate the matching accuracy and even implement an iteration process. This novel idea makes the crowdsourcing more applicable and further reduces the cost for employing dedicated people.

In Section 4, some important open issues are presented. My future work will focus on the first possible research direction, i.e., exploring techniques to be applied in crowdsourcing ER, such as correlation clustering.

6. ACKNOWLEDGMENTS

I would like to thank the China Scholarship Council to fund this research. I also express my gratitude to Eike Schallehn and Ziqiang Diao for their precious feedback.

7. REFERENCES

[1] P. Atzeni. *Database systems: concepts, languages & architectures*. Bookmantraa. com, 1999.

[2] S. B. Davidson, S. Khanna, T. Milo, and S. Roy. Using the crowd for top-k and group-by queries. In *Proceedings of the 16th International Conference on Database Theory*, pages 225–236. ACM, 2013.

[3] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 61–72. ACM, 2011.

[4] C. Gokhale, S. Das, A. Doan, J. F. Naughton, N. Rampalli, J. Shavlik, and X. Zhu. Corleone: Hands-off crowdsourcing for entity matching. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 601–612. ACM, 2014.

[5] S. Guo, A. Parameswaran, and H. Garcia-Molina. So who won?: dynamic max discovery with the crowd. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 385–396. ACM, 2012.

[6] J. Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[7] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.

[8] A. Marcus, D. Karger, S. Madden, R. Miller, and S. Oh. Counting with the crowd. *Proceedings of the VLDB Endowment*, 6(2):109–120, 2012.

[9] A. Marcus, E. Wu, D. Karger, S. Madden, and R. Miller. Human-powered sorts and joins. *Proceedings of the VLDB Endowment*, 5(1):13–24, 2011.

[10] A. Marcus, E. Wu, D. R. Karger, S. Madden, and R. C. Miller. Demonstration of qurk: a query processor for humanoperators. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 1315–1318. ACM, 2011.

[11] H. Park, H. Garcia-Molina, R. Pang, N. Polyzotis, A. Parameswaran, and J. Widom. Deco: A system for declarative crowdsourcing. *Proceedings of the VLDB Endowment*, 5(12):1990–1993, 2012.

[12] P. Venetis and H. Garcia-Molina. Quality control for comparison microtasks. In *Proceedings of the first international workshop on crowdsourcing and data mining*, pages 15–21. ACM, 2012.

[13] N. Vesdapunt, K. Bellare, and N. Dalvi. Crowdsourcing algorithms for entity resolution. *Proceedings of the VLDB Endowment*, 7(12):1071–1082, 2014.

[14] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11):1483–1494, 2012.

[15] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng. Leveraging transitive relations for crowdsourced joins. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 229–240. ACM, 2013.

[16] S. E. Whang, P. Lofgren, and H. Garcia-Molina. Question selection for crowd entity resolution. *Proceedings of the VLDB Endowment*, 6(6):349–360, 2013.