# Cost-Based and Effective Human-Machine Based Data Deduplication Model in Entity Reconciliation

Charles R. Haruna[1,2], MengShu Hou[1], Moses J. Eghan[2], Michael Y. Kpiebaareh[1], Lawrence Tandoh[1], Barbie Eghan-Yartel[2] and Maame G. Asante-Mensah[2]
[1]University of Electronic Science and Technology of China, Chengdu, China
charuna@ucc.edu.gh, mshou@uestc.edu.cn, michael.kpiebaareh@sipingsoft.com and 3065550735@qq.com
[2]University of Cape Coast, Cape Coast, Ghana
{meghan, barbie.eghan-yartel}@ucc.edu.gh and gyamfua.asantemensah@skoltech.ru

***Abstract*—In real world, databases often have several records representing the same entity and these duplicates have no common key, thus making deduplication difficult. Machine-based and crowdsourcing techniques were disjointly used in improving quality in data deduplication. Crowdsourcing were used for solving tasks that the machine-based algorithms were not good at. Though, the crowds, compared with machines, provided relatively more accurate results, both platforms were slow in execution and hence expensive to implement. In this paper, a hybrid human-machine system was proposed where machines were firstly used on the data set before the humans were further used to identify potential duplicates. We performed experiments using three benchmark datasets; paper, restaurant and product datasets. Our algorithm was compared with some existing techniques and our approach outperformed some methods by achieving a high accuracy of deduplication and good deduplication efficiency while incurring low crowdsourcing costs.**

***Index Terms*—Qualitative Error Detection, Hybrid Data Deduplication, Clustering, Pivot Graphs, Entity Resolution, Crowdsourcing.**

## I. Introduction

Entity resolution, in database, is identifying records that represent the same real-world entity [1] [2] [3] [4]. With regards to data deduplication, duplicate records are clustered into different groups. It is a major problem when records do not have a shared identifier. The objective is to find all duplicate records in table I. Though $r_1$ and $r_2$ have different entries in the product details, they are likely to represent the entity. On each record pair, the machine-based algorithm computes the similarity score between them. Pairs with their similarity values greater or equal to a given threshold are considered to be same. However, machine-based approaches often encounter challenges when processing records that highly look identical but having different entities. Furthermore, machine-based algorithms have been proved to have inferior accuracy in data deduplication. Thus, human-based methods are used to improve the accuracy of the deduplication because they are better than machines in solving complicated tasks [12].

In modern era, due to huge overheads caused by using only the crowd or the machines only, many researchers with access to crowdsourcing platforms, presented approaches on hybrid deduplication [5] [6] [10]. With a given set of data to deduplicate, machine-based algorithms are firstly used for identifying all possible pairs of records that are identical. Furthermore, on each pair, generated by the first step, the crowd is used to study them, whether they are duplicate records.

TABLE I
Sample Records of Products.

| Record ID | Product Details | Price |
|---|---|---|
| $r_1$ | HTC One Mini 16GB Wifi Black | 7200Rmb |
| $r_2$ | HTC One Mini 2 16GB Wifi White | 6000Rmb |
| $r_3$ | HTC One M8 16GB Wifi White | 4200Rmb |
| $r_4$ | HTC One M eight 2GB Waterproof White | 4900Rmb |

## II. Related Work

A survey [2] on entity resolution deduplication was presented. Also, a more larger, powerful active-learning technique to large datasets which offered probabilistic guarantees on the quality of resultant was presented in [12]. The feedback of user's verification was required on some set of candidate duplicate pairs. To establish the order in which the pairs were validated, a decision-theoretical approach was proposed. Numerous questions were generated to be posed to the community members. Resulting in the generation of a matching schema results from from the different questions, in work [13]. In crowd-clustering, records belonging to the same category was proposed in [9]. A high precision data deduplication system with a huge crowdsourcing overhead was proposed in [5]. Presented in transM, TransNode and GCER [6] [8] and [7], respectively, were methods that reduced cost but compromised data deduplication. ACD used a novel clustering algorithm, the correlation clustering, to develop a technique that offered better accuracy with moderate crowdsourcing overheads [10]. In Deco [14], with the assistance from the crowd, answering declarative queries posted over relational data, a database system was developed. In work [15], the authors proposed an SQL integration system that allowed the crowd, to use user defined functions (UDFs) to process relational databases. A data integration technique, using a hybrid machine was developed and presented in [16]. Also studies into extracting high-quality answers from crowdsourcing platforms were studied and presented [18].

978-1-7281-0120-0/18/$31.00 ©2018 IEEE

An interactive programming toolkit [20], was presented as a unified result for resolving the crowd-sourced top-k queries. The toolkit uses a top-k new confidence-aware crowdsourced algorithm, SPR. iCrowd, an adaptive crowdsourcing structure, [21], was proposed to estimate, on-the-fly accuracies of crowd. They evaluated the performances of the crowd on their tasks completed, and based on the evaluations, predicted the tasks each worker is well acquainted with.

## III. Problem Definition

The set of records and a function, be represented respectively by $R = (r_1, r_2, ..., r_n)$ and $g$, such that $r_i$ maps to the records they represent. The function $g$ usually, is often difficult to acquire, therefore, using a machine-based algorithm a similarity score function $f : R \times R \longrightarrow [0, 1]$ is assumed, and that $f(r_i, r_j)$ is equal to the possibility of $r_i$ and $r_j$ representing the entity. The goal is to attempt to split the records $R$, into a group of clusters $C = \{C1, C2, ..., Ck\}$. Therefore, for any random edge selected $(r_i, r_j)\epsilon R$, if $g(r_i)$ = $g(r_j)$ then $r_i$ and $r_j$ must be in the same cluster, implying they may be duplicates. Otherwise, they are not put in the same cluster. Similar to works [10] [25] [26], we adopt a metric cost $\Lambda(R)$, equation 1;

$$\begin{aligned}\Lambda\left(R\right) = & \sum_{r_i, r_j \epsilon R. i<j} x_{i,j} \,.\, (1 - f\left(r_i, r_j\right)) \\ + & \sum_{r_i, r_j \epsilon R. i<j} (1 - x_{i,j}\,) \,.\, (1 - f\left(r_i, r_j\right))\end{aligned} \tag{1}$$

If a record pair $r_i$ and $r_j$ belong to the same cluster, then $r_{i,j}$ = 1, else $r_{i,j}$ = 0. A penalty of $1 - f(r_i, r_j)$ is assigned on a pair that are put in the same cluster. Presented in other works [10] [27], it an is NP-hard problem minimizing the metric cost $\Lambda(R)$. In the work [10], Chromatic-correlation clustering [27], which is a variation of correlation clustering was adopted and it is a NP-hard problem.

## IV. Contributions Of The Work

The human-machine deduplication algorithm presented includes: First and foremost, to acquire candidate set $S$ which contains all pairs of records $(r_1, r_2)$ with $(r_1, r_2) \geq$ threshold, a machine-based algorithm specifically Jacccard similarity is used with a set threshold on the records $R$. Furthermore, all pairs of records in $S$ are grouped into disjoint sets $C = \{C1, C2, ..., Ck\}$. In a proposed algorithm and a HIT interface, the crowd examines and infers their results from the clusters. Finally, in the experiments, our model was compared to existing works. Our proposed approach offers a higher deduplication accuracy and incurs low crowdsourcing cost which is better than some existing work.

### A. Generating Single Cluster Algorithm

The proportion of humans that examined a record pair $(r_1, r_2)$ and concluded the pairs of the same entity is termed Crowd confidence [10]. The human-based similarity value is set as $R \times R \rightarrow [0, 1]$, such that the confidence of the crowd $f_c(r_1, r_2)$ implies that $r_1$ = $r_2$. If a pair of record was eliminated during the machine-based execution step, then $f_c(r_1, r_2)$ = 0 is set. In chromatic correlation clustering, a categorical way of clustering, clusters are structure in triangles [26]. [30] defined another similarity score called the Triangular similarity score $f_t(r_1, r_2, r_3)$ where $((r_1, r_2), (r_1, r_3), r_2, r_3))\epsilon E$, that forms a triangular cluster. It is computed by summing of the $f_c(r_1, r_2)$ in a triangular cluster, then dividing the sum by three, the edges of a triangle. Clusters are formed if $f_t(r_1, r_2, r_3) \geq$ a set threshold, implying that all the record pairs in the triangle are of the same entity.

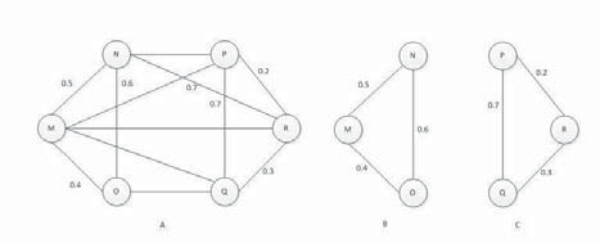

Fig. 1. An undirected graph showing crowds confidence scores.

a. In figure 1A if edge $(n, o)$ is chosen as pivot in figure 1A and all other vertices $x\epsilon R'$ which forms triangle $\langle m, n, o\rangle$, a cluster can be formed, figure 1B. The crowds confidence score of $f_c(m, o)$, 0.4 is less than the crowds confidence threshold 0.5. But the triangular similarity score $f_t(m, n, o)$ is 0.5, which is equal to the treshhold of triangular similarity score. Though one pair of records's crowd's confidence is less than the threshold, their triangular similarity score is equal to the threshold, we assume a cluster can be generated. The three records may belong to the same entity.

b. Choosing an edge $(p, r)$ in figure 1A as well as different objects $x\epsilon R'$ which forms a triangle $\langle p, q, r\rangle$, a cluster can not be formed here as shown in figure 1C. This is because the triangular similarity score $f_t(p, q, r)$, 0.4 , is lesser than the similarity score threshold 0.5. We infer that the three records are not a match.

$$\begin{aligned}\Lambda'\left(R\right) = & \sum_{r_i, r_j \epsilon R. i<j} x_{i,j} \,.\, (1 - f_c\left(r_i, r_j\right)) \\ + & \sum_{r_i, r_j \epsilon R. i<j} (1 - x_{i,j}\,) \,.\, (1 - f_c\left(r_i, r_j\right))\end{aligned} \tag{2}$$

Equation 2 states that if two records pairs $r_i$ and $r_j$ belong to a cluster, then $x_{i,j}$ = 1, else $x_{i,j}$ = 0. Using chromatic-correlation clustering, it is a NP-hard problem to minimize a metric cost $\Lambda'(R)$ in equation 2. Assuming in figure 1 clusters have already been generated, the outputs in 1B and 1C, will cost 6. This cost is the number of edges that are eliminated from a graph during execution of the algorithm. We adopt from chromatic balls algorithm [27], edge as the pivot under human-machine-based technique. A record pair ($r_i$ and $r_j$)

is chosen a pivot instead of a single object $r_i$. In chromatic balls, the construction of the clusters are categorical, and are done separately and the $i - th(i > 1)$ cluster is reliant on the first $i$ - 1 clusters, therefore the edge as pivot is non-trivial. It is time consuming and incurs large crowd overhead when generating clusters one at a time. We will further present generating multiple clusters in the work, to reduce the total time needed for completion.

*Algorithm 1:*

**Input:** an undirected graph $G = (V_r$ and $E_s)$ where $V_r \epsilon R$ and $E_s$ are the set of edges in $S$.
**Output:** A cluster set, $C$.
1: **Initialize** sets $C$ and $C_t$.
2: $R' \leftarrow R; i \leftarrow 1$.
3: **Until** there is no $(r_i, r_j) \epsilon R'$ such that $(r_i, r_j) \epsilon E$, do.
4: **Randomly** select an edge $(r_i, r_j) \epsilon R'$ as pivot, such that $(r_i, r_j) \epsilon E$.
5: **Select** all other records $r_k \epsilon R'$ for which there is a triangle $\langle r_i, r_j, r_k \rangle$.
6: **Insert** $(r_i, r_j)$, $(r_i, r_k)$ and $(r_j, r_k)$ to $C_t$
7: **Issue** to the crowd, set $C_t$
8: **For** every $E_s$ in $C_t$ do.
9: **Compute** $f_c(r_i, r_j)$.
10: **If** the crowd decides that $f_t(r_i, r_j, r_k) \geq 0.5$ then
11: **Add** $C_t$ to $C$.
12: **Delete** $G$.
13: $C(r_k) \leftarrow i$.
14: $i \leftarrow i + 1$.
15: **Return Value** $C$.
**end**

Algorithm 1, our pseudo code, which is the clustering generation phase of our work. The edges are used to generate the clusters. In this case, the algorithm chooses an edge as a pivot in each iteration. The input is an undirected graph $G = (V_r$ and $E_s)$, where $V_r \epsilon R$ and each $(r_i, r_j) \epsilon E_s$ belong to $S$. The output of the algorithm is a set of disconnected clusters $C = \{C1, C2, ..., Ck\}$. Firstly, a set $R'$ equal to $R$ is initialized, to store all the records that have not been assigned a cluster yet. An edge $(r_i, r_j)$ is selected randomly as a pivot during each iteration. With edge $(r_i, r_j)$ as pivot, and any other edges $r_k \epsilon R'$ that forms a triangle $\langle r_i, r_j, r_k \rangle$, that is $(r_i, r_j), (r_i, r_k)$ and $(r_j, r_k)$ are put in $C_t$ and is posted to a crowdsourcing platform. The crowd will examine the records in the triangle, and infers the identical pairs. The crowd's confidence $f_c(r_i, r_j)$ for all the edges are calculated in lines 9-11. Further, the triangular similarity score $f_t(r_i, r_j, r_k)$ is also calculated. If the result $f_t(r_i, r_j, r_k) \geq 0.5$, a triangular cluster is generated. The edges and records are removed from $G$ (lines 12 and 13). Otherwise if the triangular cluster can not be formed, the algorithm executes again selecting another edge. All the remaining edges and records in $R'$ forms a single clusters, lines 14 and 15. Finally, the algorithm terminates after returning a cluster $C$, with all disjoint clusters.

*1) Computational Complexity:* The computational complexity in the generation of single cluster algorithm is the selection of pivots and the clusters formation. In selecting a random pivot, a time of $O(m \log n)$ is required, where $n = |V|$ and $m = |E|$. Priority queue of edges can be built when selecting a random edge with random priorities. Whether an edge is chosen as a pivot or otherwise, it is eliminated immediately from the queue. Furthermore, in building single clusters, all the adjoin vertices of the pivot edge $(r_i, r_j)$ are accessed. These edges are thus not taking in consideration again, because when generating the next cluster, they would have been removed when the iteration terminates from the set of uncovered objects. Each edge takes $O(m)$ time to be accessed, during the selecting objects process into the current cluster, and is accessed once at most. Lastly, we deduce the complexity of algorithm 1 is $O(m \log n)$.

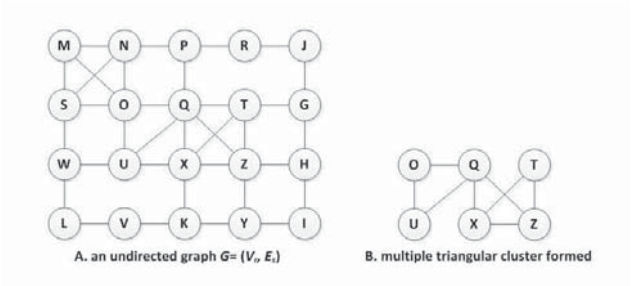


Fig. 2. An example of a Compound Clustering.

## B. Composite Clustering

In the single cluster generation algorithm 1, during each iteration, a crowd is issued with edges that has a chance of forming a cluster, to examine and infer similarity answers. The number of iterations solely determines the running time of algorithm 1. Therefore, the number of iterations can be reduced by forming, concurrently, multiple clusters by just choosing only one edge. This process will drastically reduce the running time.

Based on algorithm 1, all edges have equal chance of being chosen as a pivot in figure 2A. Some of the edges may never be issued to the crowd. Depending on the selected edge, those edges result in being removed from $G$. Executing 1 and choosing $(O, Q)$ as the pivot, triangle $\langle O, Q, U \rangle$ will be detached and issued to the crowd for examination. Based on their answers, a solitary cluster may or may not be formed. If $Q$ is removed, other potential candidates to use to generate composite clusters around may be eliminated from the graph as well.

The composite clustering attempts to reduce the probability of choosing bad edges randomly, as pivot. With vertex $r_i \epsilon R$, $d(r_i)$ represents the edges with respect to $r_i$. $\Delta(r_i)$ = max $d(r_i)$ and $\lambda(r_i)$ = arg max $d(r_i)$ are also denoted. In each iteration, a vertex $r_i$ is firstly chosen having a probability which is directly proportional to $\Delta(r_i)$. $r_i$ is the vertex having the highest number of edges attached to it. The vertex $r_j$, which is a neighbour of $r_i$ having a probability proportional to $d(r_j$, $\lambda(r_i)$, is chosen as a second vertex. Having $(r_i, r_j)$ as the edge pivot, a cluster is generated by attaching all other vertices $r_k$ so that a tiangle is formed $\langle r_i, r_j, r_k \rangle$. As long as a triangular cluster $\langle W, X, Z \rangle$, while $X$ is $r_i$ or $r_j$, and $Z$ is any other vertex that is part of current cluster can be formed, the composite clustering algorithm keeps adding $r_k$ to the cluster. The maximum number of edges of both vertices $Q$ and $X$ in figure 2 is 5. One of $Q$ and $X$ is selected as the first vertex. For instance initializing the first vertex as $r_i = X$, another vertex $r_j$ from the neighbors of $X$, with the next maximum number

of edges is chosen. $Q$ would thus be the next chosen vertex, thus $(X, Q)$ becomes the edge pivot. A temporary triangular cluster $\langle Q, X, Z\rangle$ is developed. $U$ is chosen and with $O$ and $Q$ form another triangular cluster $\langle O, Q, U\rangle$ which is joined to $\langle Q, X, Z\rangle$. Then enters $T$, and forms another triangular cluster $\langle T, X, Z\rangle$ because of $X$ and is joined to the temporary cluster of triangles $\langle O, Q, U\rangle$ and $\langle Q, X, Z\rangle$.
The temporary composite cluster generated is shown in figure 2B. The temporary cluster in figure 2B is posted to the crowd to examine the record pairs for similarities. The crowd's confidence as well as the triangular similarity scores are computed. Based on the scores, if $f_t(p, q, r)$ is lesser than the set threshold, that triangle is taken away from the cluster and reattached to $G = (V_r, E_s)$. The algorithm finally terminates by generating a solitary cluster with the remaining vertices. The solitary cluster is then issued to the crowd as well to compare for similarities.

*1) Computational Complexity:* Analogous to algorithm 1, the execution time is computational complexity under the composite clustering algorithm as well. The pivot edge choice and extra generation of different clusters determine the execution time. In determining which edge to use a pivot, a priority queue with priorities $\Delta xrnd$(a random number), is used to select the first vertex. $\Delta$ is computed for all records and this takes $O(n + m)$ time. A $O(n \log n)$ is needed when dealing with the priority queue itself. This is because, during each iteration of the algorithm, a record is only added to or removed from a queue once. To choose a the second vertex $v$, foremost, the first vertex $u$ is selected, then the neighbors of $u$ already not selected, are analyzed. For every $u$, during the whole algorithm execution process, the neighbors of $u$ are traversed only once in a time of $O(m)$. It requires a time of $O(m)$ to build the composite clusters. This is due to the fact that, it involves a time of $O(1)$ to access every edge during each traversal of the graph. In conclusion, the computational complexity of the composite clustering algorithm is $O(n(\log n) + m)$.

## V. Experiments

In the experiments section,with regards to accuracy of deduplication, crowdsourcing overheads, and the efficiency of the proposed model, experiments were performed and results evaluated against Transnode [8], TransM [6], CrowdEr [5] and ACD [10].

### A. Experiment Setup

Three benchmark datasets namely Paper [28], Product [29] and Restaurant [17] also used in some hybrid deduplication techniques [5] [6] [7] were employed. II shows a summary of the number of records and entities in all three datasets. Sorted neighbourhood clustering algorithm [7] was used on the crowd answers to build the clusters in crowdEr and was thus denoted as $CrowdEr^*$ in our experiments. To balance deduplication accuracy, the F-1 measure was used on all approaches as well as in this work. In addition, the cost of the methods used were evaluated in terms of the number of pairs of records, which the crowd incurred. The Jaccard similarity function was used to compute the machine-based scores to form the set $S$ and the threshold value, $t$, was set at 0.4. Only record pairs having similarity scores greater than $t$ are grouped as set $S$ shown in table II. Motivated by CrowdEr [5] and ACD [10], we contacted the authors for their answers they used in their experiments and reuse in our experimental evaluation.

### B. Evaluation of our Work With Existing Hybrid Data Deduplication Methods

Experimantal results of our work was compared with some methods based on the following;

1. Using the F1-measure on Deduplication accuracy.
2. Efficiency of the Crowdsourcing is evaluated by determining the number of iterations record pairs $(r_i, r_j)$ are examined by the crowd.
3. Overhead of Crowdsourcing which is the cost incurred crowdsourcing the pairs of records.

**Accuracy of Deduplication:**
Figure 3, shows results of experiments performed employing the F-1 measure against each of the existing methods. F-1 measure is calculated as the number of pairs of records that were crowdsourced. Under $3w$ and $5w$ settings, deduplication accuracy experiments were performed on all three datasets. Under both settings on the three datasets, $CrowdEr^*$ had the highest accuracy, ACD followed with the second highest. The third highest was our algorithm, thus performed better than TransNode and TransM. Under both settings, TransNode had the least accuracy followed by TransM on all datasets. TransM and TransNode performed poorly especially on paper dataset test. But on both product and restaurant, they significanlty performed better. Compared to ACD and $CrowdEr^*$, our algorithm showed almost equal accuracy, with only slight differences in the results. Under both settings, using datasets Product and Restaurant, the techniques compared, had fairly equal accuracy.

**Overheads of Crowdsourcing:**
Under $3w$ and $5w$ settings, as shown and described earlier in fig. 3, $CrowdEr^*$ provided better accuracy, but it incurs huge crowdsourcing cost especially on Paper dataset, fig. 4. Similar to the data deduplication accuracy graph in fig. 3, our algorithm is almost the same as ACD under crowdsourcing overheads as well, but incuured little crowdsourcing cost than ACD. On both paper and Product under $3w$ and $5w$, TransNode and TransM were costly than the cost of our algorithm, ACD and $CrowdEr^*$. On the other hand, on Restaurant dataset and under $3w$ and $5w$, evaluations against our technique and ACD, TransNode and TransM incurred more cost.

**Efficiency of Crowdsourcing:**
The evaluation on crowdsourcing efficiency of all methods is shown in fig. 5. The number of crowd iterations incurred are shown in using the graphs. $CrowdEr^*$ executes needing the

TABLE II
FEATURES OF DATASETS AND CROWD'S ANSWERS.

| Datasets | # of Records | # of Entities | # of $S$ | Rate of Errors of Crowds $(3w)$ | Rate of Errors of Crowds $(5w)$ |
|---|---|---|---|---|---|
| Paper | 997 | 191 | 29,581 | 23% | 21% |
| Product | 858 | 752 | 4,788 | 0.8% | 0.2% |
| Restaurant | 3,073 | 1,076 | 3,154 | 9% | 5% |

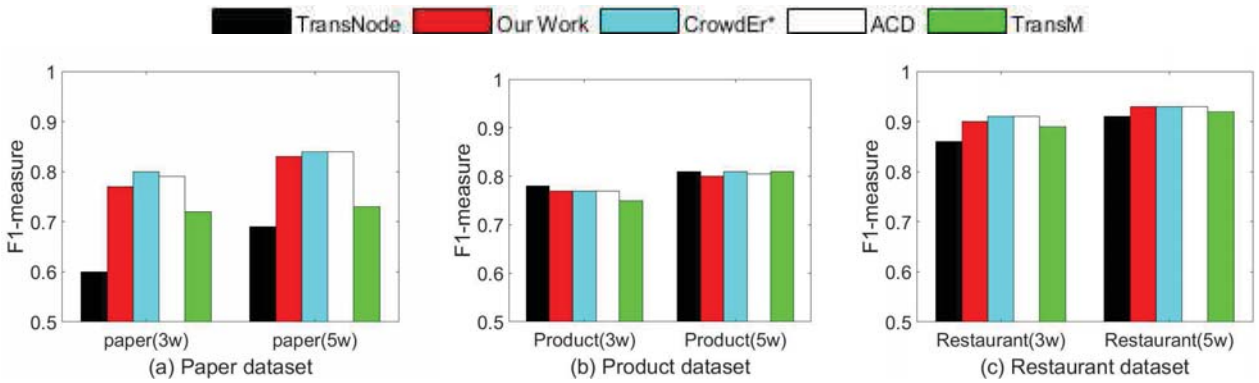


Fig. 3. Evaluation of deduplication accuracy.

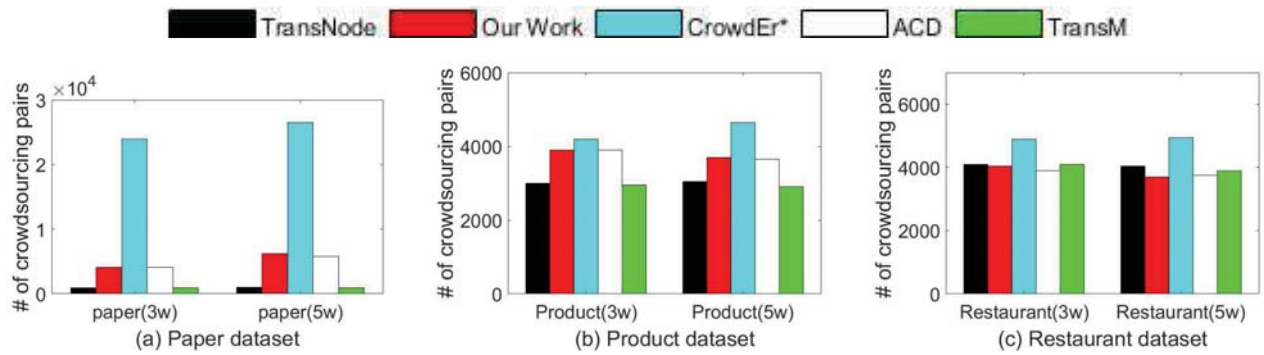


Fig. 4. Evaluation of crowdsourcing overheads.

very least number of iterations, that is less than two (2). It is evident that the other methods are all almost equal on the three datasets, performed under both $3w$ and $5w$. The approach in this work had a better efficiency than TransM and $CrowdEr^*$. While of all the methods, ACD had the better efficiency than to our work.

From the results of the experiments and evaluations made, we can conclude that Algorithm 1 provides a data deduplication accuracy which is fairly higher and with relatively low crowdsourcing overheads compared to the other works used in the experiments. Finally, the crowdsourcing efficiency is almost on the same level as the existing works. Therefore, the hybrid deduplication system can be deemed good to implement.

## VI. CONCLUSION

A hybrid deduplication method using a Jaccard similarity and a chromatic correlation clustering which has not been used in the research under entity reconciliation, has been proposed. In the experiments, comparing this work with Transnode, TransM, CrowdEr and ACD, the efficiency, and accuracy of the deduplication were greatly improved. Finally, our algorithm incurred minimum crowdsourcing overheads compared to almost all of the work but CrowdEr.

## ACKNOWLEDGMENT

We appreciate the comments provided by Glenn Joseph Haruna that helped to improve our work. We are most grateful.

## REFERENCES

[1] Christen, P.: Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Data-Centric Systems and Applications. Springer, Berlin (2012)
[2] Elmagarmid,A.K., Ipeirotis, P.G.,Verykios,V.S.: Duplicate record detection: A survey. IEEE Trans. Knowl. Data Eng. 19(1), 116 (2007)
[3] Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Comput. Surv. 31(3), 264323 (1999)
[4] Winkler,W.:Overviewof record linkage and current research directions. Tech. rep., Statistical Research Division, U.S. Bureau of the Census, Washington, DC (2006)
[5] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. Proceedings of the VLDB Endowment, 5(11):14831494, 2012.

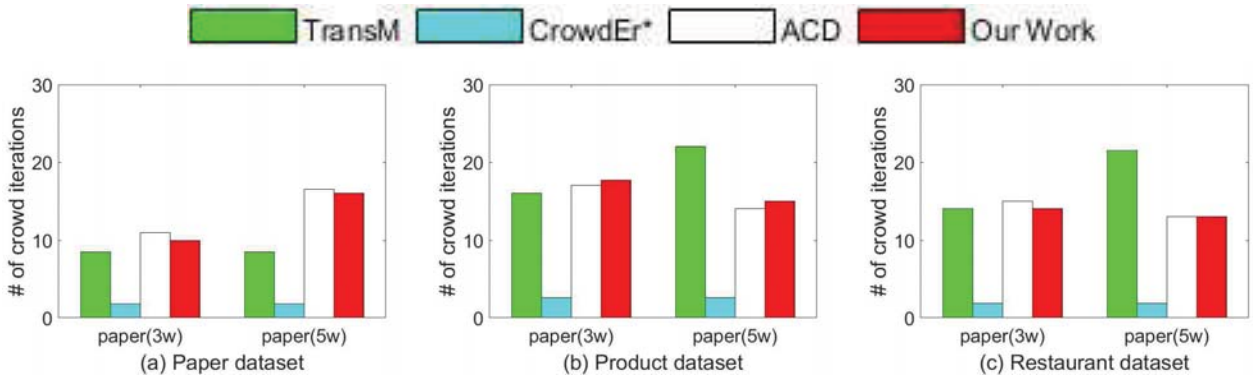


Fig. 5. Evaluation of crowdsourcing efficiency.


[6] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng. Leveraging transitive relations for crowdsourced joins. In Proceedings of the 2013 international conference on Management of data, pages 229240. ACM, 2013.

[7] S. E. Whang, P. Lofgren, and H. Garcia-Molina. Question selection for crowd entity resolution. Proceedings of the VLDB Endowment, 6(6):349360, 2013.

[8] N. Vesdapunt, K. Bellare, and N. Dalvi. Crowdsourcing algorithms for entity resolution. Proceedings of the VLDB Endowment, 7(12), 2014.

[9] R. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. In NIPS, pages 558566, 2011.

[10] Wang, S., Xiao, X. and Lee, C.H., 2015, May. Crowd-based deduplication: An adaptive approach. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (pp. 1263-1277). ACM.

[11] H. Kopcke, A. Thor, and E. Rahm. Evaluation of entity resolution approaches on real-world match problems. PVLDB, 3(1):484493, 2010.

[12] A. Arasu, M. Gotz, and R. Kaushik. On active learning of record matching packages. In SIGMOD Conference, pages 783794, 2010.

[13] R. McCann, W. Shen, and A. Doan. Matching schemas in online communities: A web 2.0 approach. In ICDE, pages 110119, 2008.

[14] A. Parameswaran, H. Park, H. Garcia-Molina, N. Polyzotis, and J. Widom. Deco: Declarative crowdsourcing. Technical report, Stanford University. http://ilpubs.stanford.edu:8090/1015/.

[15] A. Marcus, E. Wu, D. R. Karger, S. Madden, and R. C. Miller. Human-powered sorts and joins. PVLDB, 5(1):1324, 2011.

[16] S. R. Jeffery, L. Sun, M. DeLand, N. Pendar, R. Barber, and A. Galdi. Arnold: Declarative crowd-machine data integration. In CIDR, 2013. Proceedings of the ACM SIGKDD workshop on human computation, pages 6467. ACM, 2010.

[17] http://dbs.uni-leipzig.de/Abt-Buy.zip.

[18] A. G. Parameswaran, H. Garcia-Molina, H. Park, N. Polyzotis, A. Ramesh, and J. Widom. Crowdscreen: Algorithms for filtering data with humans. In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pages 361372. ACM, 2012.

[19] P. Wais, S. Lingamneni, D. Cook, J. Fennell, B. Goldenberg, D. Lubarov, D. Marin, and H. Simons. Towards building a high-quality workforce with mechanical turk. Proceedings of computational social science and the wisdom of crowds (NIPS), pages 15, 2010.

[20] Li, Y., Kou, N.M., Wang, H. and Gong, Z., 2017. A confidence-aware top-k query processing toolkit on crowdsourcing. Proceedings of the VLDB Endowment, 10(12), pp.1909-1912.

[21] Fan, J., Li, G., Ooi, B.C., Tan, K.L. and Feng, J., 2015, May. icrowd: An adaptive crowdsourcing framework. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (pp. 1015-1030). ACM.

[22] Xiao, C., Wang, W., Lin, X., Yu, J.X. and Wang, G., 2011. Efficient similarity joins for near-duplicate detection. ACM Transactions on Database Systems (TODS), 36(3), p.15.

[23] R. J. Bayardo, Y. Ma, and R. Srikant. Scaling up all pairs similarity search. In WWW, pages 131140, 2007.

[24] S. Chaudhuri, V. Ganti, and R. Kaushik. A primitive operator for similarity joins in data cleaning. In ICDE, page 5, 2006.

[25] O. Hassanzadeh, F. Chiang, H. C. Lee, and R. J. Miller. Framework for evaluating clustering algorithms in duplicate detection. Proceedings of the VLDB Endowment, 2(1):12821293, 2009.

[26] Z. Chen, D. V. Kalashnikov, and S. Mehrotra. Exploiting context analysis for combining multiple entity resolution systems. In Proceedings of the 2009 ACM SIGMOD Internation

[27] Bonchi, F., Gionis, A., Gullo, F., Tsourakakis, C.E. and Ukkonen, A., 2015. Chromatic correlation clustering. ACM Transactions on Knowledge Discovery from Data (TKDD), 9(4), p.34.

[28] http://www.cs.umass.edu/ mccallum/data/cora-refs.tar.gz.

[29] http://www.cs.utexas.edu/users/ml/riddle/data/restaurant.tar.gz.

[30] Haruna, C.R., Hou, M., Eghan, M.J., Kpiebaareh, M.Y. and Tandoh, L., 2018, November. A Hybrid Data Deduplication Approach in Entity Resolution Using Chromatic Correlation Clustering. In International Conference on Frontiers in Cyber Security (pp. 153-167). Springer, Singapore.