# A pruning algorithm for Meta-blocking based on cumulative weight

**Fulin Zhang**[1], **Zhipeng Gao**[1] **and Kun Niu**[2]

[1] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2] School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
Email: zhangfulin@bupt.edu.cn

**Abstract.** Entity Resolution is an important process in data cleaning and data integration. It usually employs a blocking method to avoid the quadratic complexity work when scales to large data sets. Meta-blocking can perform better in the context of highly heterogeneous information spaces. Yet, its precision and efficiency still have room to improve. In this paper, we present a new pruning algorithm for Meta-Blocking. It can achieve a higher precision than the existing WEP algorithm at a small cost of recall. In addition, can reduce the runtime of the blocking process. We evaluate our proposed method over five real-world data sets.

## 1. Introduction

Entity Resolution(ER) first proposed by [1]. ER has a significant application in data cleaning and data integration. As the amount of the global data grows exponentially, it has been a hot topic. It is the task to determine whether two different records in one or more data sources describe the same real world object, sometimes referred to as entity matching, record linkage or object matching. ER needs to compute the similarity between records to detect the duplicates. Due to its characteristics (one record has to compared with all others), it is very expensive for large data sets especially the rapid development of the Internet causes a fast growth of the amount of entity. Assuming that the size of two data sets that need to be matched is m and n respectively, m×n comparisons required if we do it in a brute-force way. The consumption of computing resources is extremely large, and it will seriously affect the efficiency. It typically employs a filter process commonly called blocking before calculation which leaving the most similarity records in the same block, such that the comparisons can be reduced.

## 2. Related Work

Blocking is a crucial step for ER. It can improve the efficiency of the ER process. The earliest blocking method called Traditional Blocking [2], each record can only be placed into one block and records with the same blocking key values (BKV) [3] are placed in the same block. It can be easily implemented by inverted index. The Sorted Neighbourhood indexing [4] method sorts the records according to the BKV and then moves a sliding window of the size fixed to on the sorted data set, comparing the records in a window. The main disadvantage is that the size of the sliding window. Assuming that two similar records are not so close that they are not in the same window, and then they will be unmatched. [5] Proposed a method that can change the size of the sliding window dynamically. For the data sets that contains a large number of errors, the above two methods cannot perform well because the matching records are often placed into different blocks. Q-gram [6] based blocking inserts records into more than one block to make up for the above shortcomings. Its BKVs is the substring of lengths q. The Canopy clustering algorithm [7-8] uses a rough distance calculation method to allocate data into different overlapping

subsets in the first stage, and then only calculates records in the same overlapping subset to reduce the number of records need for calculation. [10] Proposed a pay-as-you-go method for ER. It can efficiently process the ER with a limited resource and time (real-time system) by using hints. [12] Proposed an iterative blocking framework, which exploits the results of processed blocks to save many comparisons. It can achieve a higher accuracy. These methods above are not suitable for highly heterogeneous information spaces (HHIS) [11] where there have a lot of noise, no structural data, and the amount of data is large. In addition, they all need a predefined schema. Meta-Blocking [13], a schema-agnostic method, achieves a good balance between precision and recall. Its core idea is the notion of blocking graph, which can discard the redundant and superfluous comparisons by a pruning algorithm. It has four steps: (i) build the blocking graph from the input block collection. (ii) Then use weighting schemes to set edge weight. (iii) Pruning the blocking graph use edge-centric algorithms or node-centric algorithms. (vi) Transform the pruned blocking graph into a new block collection. It takes the entity resolution to the next level.

The contributions of our work are as follows: (1) A new proposed pruning algorithm for Meta-blocking that can raise precision at a small cost in recall and it is more efficient than the existing WEP algorithm. (2) A series of thorough experiments to verify our proposed method over five real-world data sets.

The remainder of this paper is organized as follows: Section 3 describes the main notions of the work. Section 4 propose our new pruning algorithm. Section 5 presents our experiments and the results of evaluation. Section 6 contains conclusions of the paper and some recommendations for the future work.

## 3. Preliminaries

In this section, we introduce some main notion of this work.

Entity Profile. An entity profile is something with distinct and real existence. It can be defined as follows: $e_i$ = {id, $name_{i1}$, $value_{i1}$, $name_{i2}$, $value_{i2}$…}, id is a globally unique flag; $name_{ij}$ and $value_{ij}$ are the names and values of an attribute that the entity has.

Entity Resolution(ER). The ER process is illustrated in figure1.It commonly involves two steps.

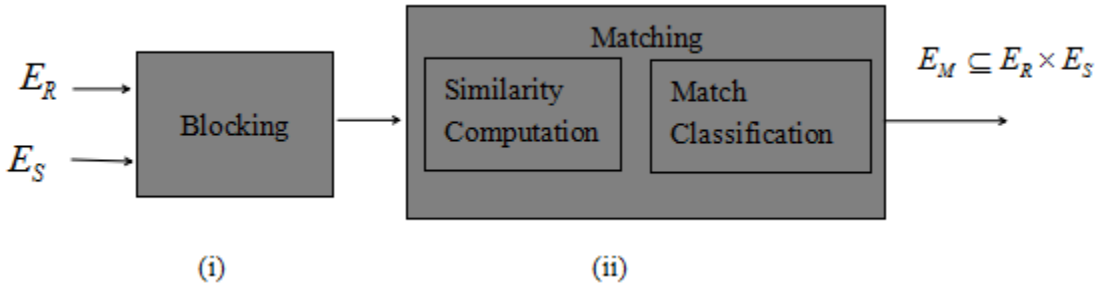



**Figure 1.** The traditional ER process

$E_R$, $E_S$ is a set of entity profiles. (i) Blocking to reduce the total comparisons. Group similar records within blocks based on blocking key. (ii) Entities in the same block use a similarity function to Compared with each other. EM is the detected records that refer to the same real world object.

Clean-Clean ER. both $E_R$ and $E_S$ themselves do not contain duplicates. Dirty-Dirty ER: both $E_R$ and $E_S$ themselves contain duplicates.

Blocking. ER is an intrinsically quadratic task, the total number of record comparisons equals to $|M||N|$ , with $|\cdot|$denoting the number of records in a data set. This is the bottleneck when ER scales to large entity collections. To reduce the large amount of comparisons, some blocking techniques are employed. Blocking can put the most similar entities together, and only candidate entity profiles in the same block are compared.

Meta-blocking. It is a schema-agnostic method that can reduce the redundant and superfluous comparisons by rebuild the block collection. It can improve the efficiency at a minor cost in effectiveness. The figure 2 is ER process with meta-blocking. We can note that it does not replace the

existing blocking methods. In contrast, it can combine with the existing methods to make ER process better.

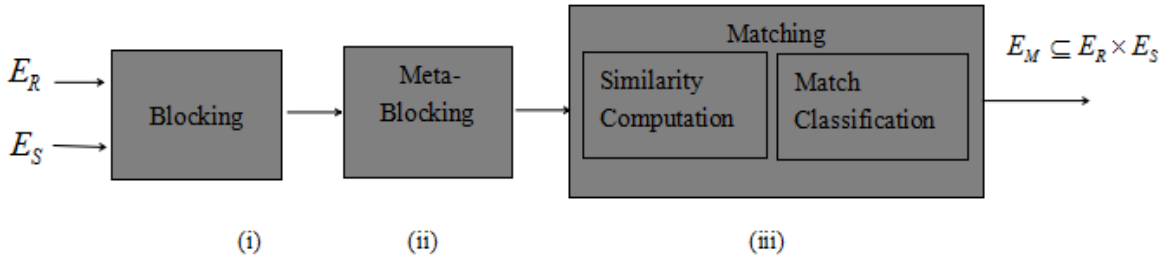



**Figure 2.** The ER process with Meta-blocking.

In our experiments, we use the following measures to evaluate our method. Pairs Quality(PQ): Estimate the proportion of non-redundant comparisons in the block. It is corresponding to precision used in the field of information retrieval. Its value between 0 to 1, A higher the value means a higher efficiency of the block. Pairs Completeness (PC): Estimate the proportion of the detected duplicates. It is corresponding to recall used in the field of information retrieval. Its value between 0 to 1, A higher value means a higher effectiveness of the block. Reduction Ratio (RR): Estimate the proportion of saved comparisons by blocking. Its value between 0 to 1, the higher value means the higher efficiency of the block.

## 4. Our Proposed Method

We introduce a new pruning algorithm Cumulative Weight Edge Pruning (CWEP) in this section. Differ from WEP's minimum edge weight and CEP's top K edges with the maximum weight [13], our method focuses on the cumulative weight. As is showed in the outline of Algorithm, the input of the algorithm is a blocking graph and a global thresholdθ. The blocking graph was built by block collections with edges weighted by different weighting schema. The output is an undirected pruned blocking graph. The algorithm first put graph's all edges into a sorted stack (line 1-3). The edges in the sorted stack are in descending order of weights. In addition, we can calculate the SumWeight of all edges. Then the algorithm removes edges from the sorted stack until the sum weight of edges in the sorted stack less than θ*SumWeight (line 4-5). Finally, the algorithm iterates again to remove edges that are not contained in the sorted stack (line 6-9). The remained edges in the sorted stack can form the new graph. The time complexity of our algorithm is O (|B|). |B| is the cardinality of the original block collections. In our work, we set θ to 0.5.

**Algorithm :Cumulative Weight Edge Pruning(CWEP)**

**Input**: (i) $G_B^{in}$ the blocking graph (ii) θ the global cumulative weight pruning criterion
**Output**: $G_B^{out}$ the undirected pruned blocking graph

1 SortedStack←{};
2 **foreach** $\boldsymbol{e_{i,j}} \in \boldsymbol{E_B}$ **do**
3 SortedStack.push();
4 **while** Cumulative Weight>θ*SumWeight **do**
5 SortedStack.pop();
6 **foreach** $\boldsymbol{e_{i,j}} \in \boldsymbol{E_B}$ **do**
7 **if** $e_{i,j} \notin$ SortedStack **then**
8 $\boldsymbol{E_B} \leftarrow \boldsymbol{E_B} - \{\boldsymbol{e_{i,j}}\}$
9 **return** $G_G^{out} = \{V_B, E_B, WS\}$

## 5. Experiment

In this section, we do a series of experiments to demonstrate the performance of our method CWEP.

*5.1. Setup*

Our approach was implemented in Java 1.8. Our experiments were performed on a server with Intel(R) Core(TM) i7-4790 CPU 3.60GHz and 16.0GB of RAM. The operating system is Windows 7 Professional. The data sets we used in our experiments are showed in table 1.

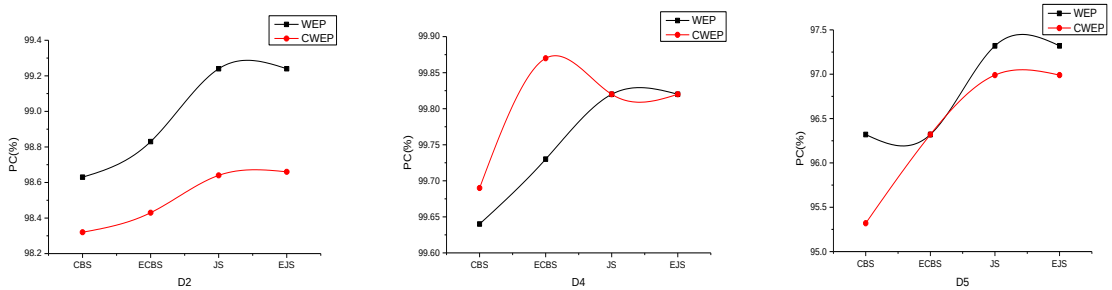
**Table 1.** The datasets we used in our experiments.

| | Entities | Name-Value pairs | Existing duplicates | Brute-force Comparisons |
|---|---|---|---|---|
| D1 | 1,076/1,076 | 2,568/2,308 | 1,076 | 1,157,776 |
| D2 | 1,295 | 7,166 | 17,184 | 837,865 |
| D3 | 4,910 | 19,626 | 2,224 | 12,051,595 |
| D4 | 9,763 | 183,072 | 299 | 95,525,976 |
| D5 | 1,354/3,039 | 1,354/3,039 | 1,104 | 4,114,806 |

D1 and D5 are clean-clean datasets, the others are dirty datasets. D1 is the "Abt_Buy" dataset containing 1076 product records of the Abt website and 1076 product records form the Buy site. D2 is the "Cora" dataset containing 1295 records of machine learning publications. D3 is the "ACM" dataset containing 4910 records from Association for Computing Machinery. D4 is the "Cddb" dataset containing 9763 records. It is a database containing audio CD information. D5 is the "Amazon-Google Product" dataset. It contains product records from amazon and google.

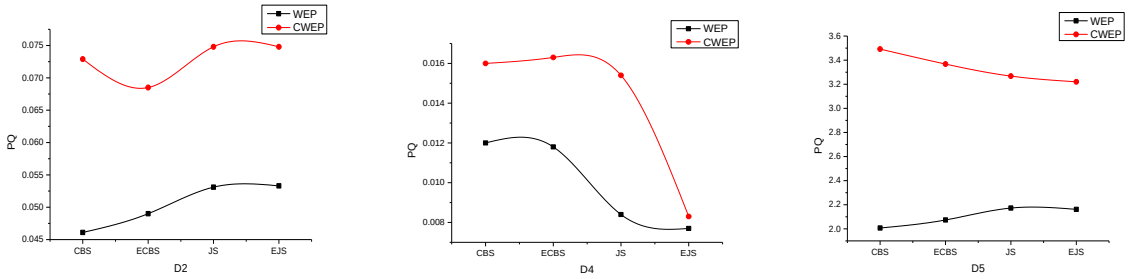
*5.2. Discussion*

The results are showed in the following figures. The x-axes are the different weighting schemas proposed by [13]. Figure 3(a) to 3(c) are the PC over different datasets, and figure 4(a) to 4(c) are the PQ (Due to the lack of space, we omit the other two data sets). We can observe that the interval between two lines in 3(a) to 3(c) is smaller than in 4(a) to 4(c) (except D4). It means that our CWEP algorithm can achieve a higher accuracy than WEP at a small cost of recall. It can increase precision by an average of 20 percent than WEP. In data set 2, it even reaches to 40 percent. Figure 5(a) to 5(c) are the RR over different datasets. The RR of CWEP is also higher than WEP (except D4). It means our algorithm have a higher efficiency than WEP.



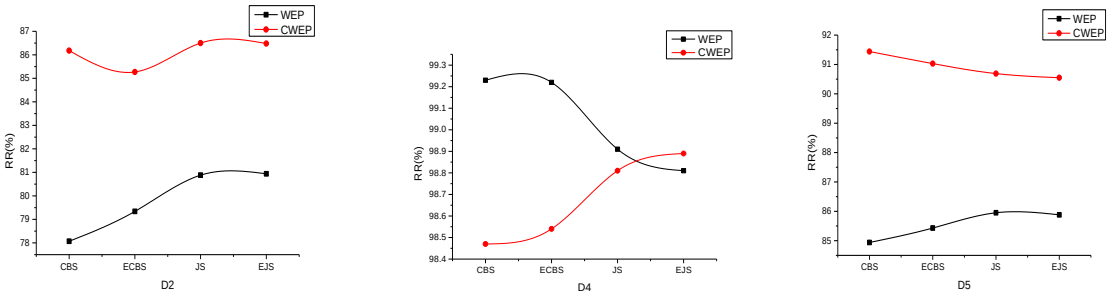

(a) PC over D2 (b) PC over D4 (c) PC over D5

**Figure 3.** Comparing effectiveness between CEP and CWEP across three datasets




(a) PQ over D2 (b) PQ over D4 (c) PQ over D5

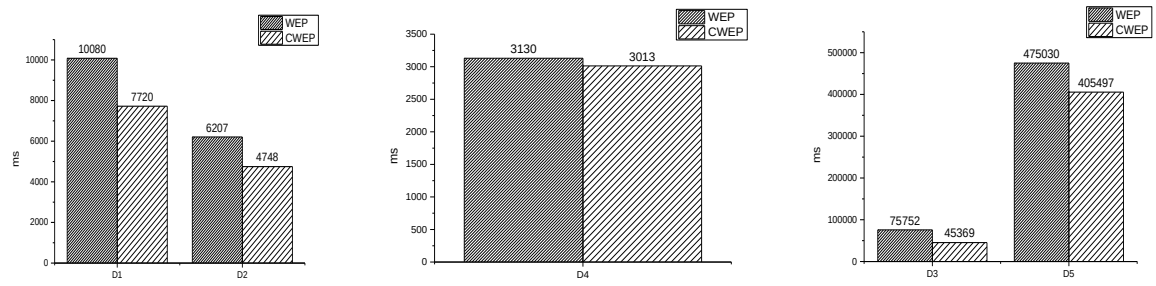
**Figure 4.** Comparing efficiency between CEP and CWEP across three datasets




(a) RR over D2 (b) RR over D4 (c) RR over D5

**Figure 5.** Comparing RR between CEP and CWEP across three datasets

We also do the sensitivity analysis experiment. We set the pruning threshold θ from 0.1 to 0.9, we note that the higher the threshold, the lower the RR and the higher the PC in our experiment. It means our algorithm is robust. The total runtime of WEP and CWEP over different weighting schemas are showed in figure 6(a) to (c). As expected, our CWEP algorithm is faster than the WEP algorithm. Except for D4, the CWEP can save more than 17 percent runtime.




(a) (b) (c)

**Figure 6.** Comparing runtime of WEP and CWEP across all datasets

## 6. Conclusion

In this paper, we introduced a new pruning algorithm CWEP for Meta-blocking. We evaluated our algorithm over five real-world datasets. The result showed that our algorithm could achieve a higher accuracy than the existing WEP algorithm. In addition, it is more efficient. In the future, we plan to enhance its efficiency and apply our algorithm to the Parallel Meta-blocking.

## 7. Acknowledgments

This work is supported by the National Natural Science Foundation of China (61272515, 61372108), and National Science & Technology Pillar Program (2015BAH03F02).

## 8. References

[1] Newcombe HB, Kennedy JM, Axford SJ and James AP. "Automatic Linkage of Vital Records." 1959 Science 130.3381:954.
[2] P. Fellegi I and B. Sunter A. "A Theory for Record Linkage." 1969 Journal of the American Statistical Association 64.328:1183-210.
[3] Christen P. "A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication." 2012 IEEE Transactions on Knowledge & Data Engineering 24.9:1537-55.
[4] Mauricio A. Hernández and Stolfo SJ. "The merge/purge problem for large databases." 1995 Acm Sigmod International Conference on Management of Data ACM: 127-38.
[5] Yan S, Lee D, Kan MY and Giles LC. "Adaptive sorted neighborhood methods for efficient record linkage." 2007 Acm/ieee-Cs Joint Conference on Digital Libraries ACM: 185-94.
[6] Baxter R, Christen P and Epidemiology CF. "A Comparison of Fast Blocking Methods for Record." 2003 Kdd Workshops: 25--seven.
[7] Richman J and Richman J. "Learning to match and cluster large high-dimensional data sets for data integration." 2002Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ACM: 475-80.
[8] Mccallum A, Nigam K and Ungar LH. "Efficient clustering of high-dimensional data sets with application to reference matching." 2000 International Conference on Knowledge Discovery and Data Mining DBLP: 169-78.
[9] Faloutsos, C and Lin KI. "FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets." 1995 ACM SIGMOD Record 24.2:163-74.
[10] Whang SE, Marmaros D and Garcia-Molina H. "Pay-As-You-Go Entity Resolution." 2013 IEEE Transactions on Knowledge & Data Engineering 25.5:1111-24.
[11] Papadakis G, Ioannou E, Palpanas T, Niederee C and Nejdl W. "A blocking framework for entity resolution in highly heterogeneous information spaces." 2013 IEEE Transactions on Knowledge and Data Engineering 25.12: 2665-82.
[12] Whang SE, Menestrina D, Koutrika G, Theobald M and Garcia-Molina H. "Entity resolution with iterative blocking." ACM: 219-32.
[13] Papadakis G, Koutrika G, Palpanas T and Nejdl W. "Meta-Blocking: Taking Entity Resolution to the Next Level." 2014 IEEE Transactions on Knowledge & Data Engineering 26.8:1946-60.