CTextEM: Employing Compound Textual Information in Entity Matching

Qiang Yang, Zhixu Li *, Binbin Gu, An Liu, Pengpeng Zhao, Guanfeng Liu, and Lei Zhao

School of Computer Science and Technology, Soochow University, China
{qiangyanghm,gu.binbin}@hotmail.com,{zhixuli,ppzhao,gfliu}@suda.edu.cn

Abstract. Entity Matching (EM) identifies records referring to the same entity within or across databases. Existing EM methods measuring the similarities between structured attribute values (such as digital, date or short string values) may fail when the structured information is not enough to reflect the matching relationships between records. Nowadays more and more data sets have some unstructured textual attribute containing extra compound textual information (or what we call as CText) of the record, but seldom work has been done on using the information for EM. Conventional string similarity metrics such as edit distance or bag-of-words are unsuitable for measuring the similarities between CTexts since there are hundreds or thousands of words with each CText, while existing topic models either can not work well since there is no obvious gaps between the various topics in CText. In this paper, we work on employing CText in EM. We not only propose a novel cooccurrence-based topic model to identify various topics of each CText such that to measure the similarity between CTexts on multiple topic dimensions, but also find ways to decrease the high cost of employing CText in EM from $O(n+\frac{e}{2})$ to $O(n+\frac{e}{2})$. Our empirical study shows that our method outperforms several previous methods and baselines on reaching a higher EM precision & recall, and can greatly improve the EM efficiency by more than 60% on several real data collections.

1 Introduction

As the data explosion for decades, the redundancy and inconsistency between records becomes more and more serious within and across databases. Entity Matching (EM), also known as record linkage or duplicate detection, aims at finding out records referring to the same entity within or across relation tables.

So far, plenty of work has been done on EM according to the similarities [15] or correlations [20] between various kinds of structured attribute values of the records such as digital values, date values or short string values (see [10] for a survey). However, EM based on structured information only may easily fail for lack of enough information. For instance, nowadays we have various kinds of second-hand goods (like cars, houses, or furnitures) for selling online, but there

* The corresponding author

	Residence Community	Location (District)	Type	Size	Floor	General Supplimental Description
r_1	Eastern District Court	Canglang-Xujiang	Residence	75 m^2	3/15	1. Community Planning and unique warmth, flowers and trees patchwork, like a garden, world without dispute, furniture and appliances equipped well. 2. Hardcover, well-groomed, color matching gentle, facing south.
r_2	Eastern District Court	Canglang-Xujiang	Residence	75 m^2	3/15	1. Community Planning and unique warmth, flowers and trees patchwork, furniture and appliances equipped well. 2. Hardcover, color matching gentle, facing south
r_3	Eastern District Court	Canglang-Xujiang	Residence	-	3/15	1. Community Planning and unique warmth, flowers and trees patchwork, without dispute, furniture and appliances equipped well. 2. Hardcover, well-groomed, color matching gentle , facing south
r_4	Oak Bay Garden	Xiangcheng-Yuanhe	Apartment	100 m^2	25/29	1. Hardcover, south, nice view, good lighting, air conditioning, water heaters, washing machines, refrigerators, closed kitchen and other necessities, 2. free of parking, free of property charges, bag check
r_5	Eastern District Court	Canglang-Xujiang	Residence	75 m^2	3/15	1. Unique warmth, community planning well, flowers and trees patchwork, furniture and appliances equipped well. 2. Hardcover, relaxing at ease, world without dispute, color matching gentle, facing south.
r_6	Eastern District Court	Canglang-Xujiang	Residence	75 m^2	3/15	1. Community Planning and unique warmth, flowers and trees patchwork. 2. Hardcover, furniture and appliances equipped well, color matching gentle, facing east
r_7	Oak Bay Garden	Xiangcheng-Yuanhe	Apartment	100 m^2	25/29	1. Hardcover, south, good lighting, air conditioning, water heaters, washing machines, refrigerators, closed kitchen and other necessities, free of property charges, bag check
r_8	Oak Bay Garden	Xiangcheng-Yuanhe	Apartment	100 m^2	-	1. Hardcover, south, nice view, air conditioning, water heaters, washing machines, refrigerators, closed kitchen and other necessities 2.free of parking, bag check

Table 1. Example "House Renting Information" Table with CTexts, in which r_1, r_2, r_3, and r_5 refer to the same house, and r_4, r_7 and r_8 are the another same house

might be only limited structured information about the good such as those shown in Table 1. Relying on the structured information only, sometimes we can not identifying records referring to the same entity.

But on the other hand, there are usually some long free-text description which contains Compound Textual Information (or **CText** for short) about each database record. For example in Table 1, the values under the "General Supplemental Description" attribute contain some extra information such as "orientation", "virescence", "type of decoration" and so forth. However, conventional string similarity metrics such as edit distance or bag-of-words are unsuitable for measuring the similarities between CTexts since there are usually hundreds or even thousands of words with each CText where much noisy information is mixed with useful information.

There have been some efforts on using CText for EM. For instance, Ektefa et. al. [8] calculate both a string similarity score and a semantic similarity score between CTexts. However, the string similarity is simply calculated by Jaccard and the semantic similarity is simply defined by several general "fields" (such

as Address, City, Phone, Type) in the WordNet, which only works well on some specific data sets. Gao et. al. [11] put forward a semantic features based method, which defines a semantic feature vector like {*time, location, agentive, objective, activity*} for every CText, and then train a classifier to identify duplicate records based on their feature vectors. However, this method is also limited in the dimensions of the features they employed, and thus can not be easily applied to the other data sets. In addition, the existing topic models such as *Latent Dirichlet Allocation(LDA)* [3], *Latent Semantic Analysis(LSA)* [16] and *Probabilistic Latent Semantic Analysis(PLSA)* [13] could identify topics from free texts such as the topics of news like "education", "financial", "sports" or "music" etc. However, as a general description/metadata about a record, the topics in an CText can be seen as sub-topics of a general topic, thus they share many topic words and there is no clear gap between these topics. On the other hand, a topic in CText can be very short (like several words), thus we can hardly learn any topic words as we could do with previous topic models.

In this paper, we propose a novel model based on the co-occurrences between phrases to identify topics from a CText, such that we can better utilize the topic-based information for EM. Specifically, We firstly count up the rate of co-occurrence among phrases to establish a topic vector model on the training data, and then we construct comparison vectors of records based on topic vector model to measure the similarity between CTexts on multiple topics. We finally compute the similarity of entity pairs. In this way, we can acquire the fine-grained information of characteristic for CText. On the one hand, the accuracy of EM can be improved to some extent because of definite comparison objects. On the other hand, the efficiency of EM can be improved due to decreasing the time cost for different characteristic.

Besides accuracy, the efficiency of EM is also an important issue, especially for EM with CText. To decrease the overhead of EM, we consider to employ structured data to decrease the comparison times between records with an efficient blocking algorithm. The blocking algorithm is an approach to build up a kind of Hash Sequence so as to reduce the number of comparison.

We summarize our contributions as follows:

- We work on a novel problem that using CText information for EM.
- We put forward an algorithm to acquire topic-based information from CText, such that we can reach a higher precision and recall of EM.
- We propose to use structured information together with CText information to improve the efficiency of the algorithm.

We experimentally verify the effectiveness and scalability of our two novel algorithms. We find that our baseline algorithm can improve precision greatly but bad effectiveness, while our improved algorithm can cope with the records with CText with high accuracy and effectiveness.

Roadmap. The rest of the paper is organized as follows: We give the workflow overview and then define the problem in Sec. 2, and then present our algorithm in Sec. 3. After reporting our experimental study in Sec. 4, we cover the related work in Sec. 5. We finally conclude in Sec. 6.

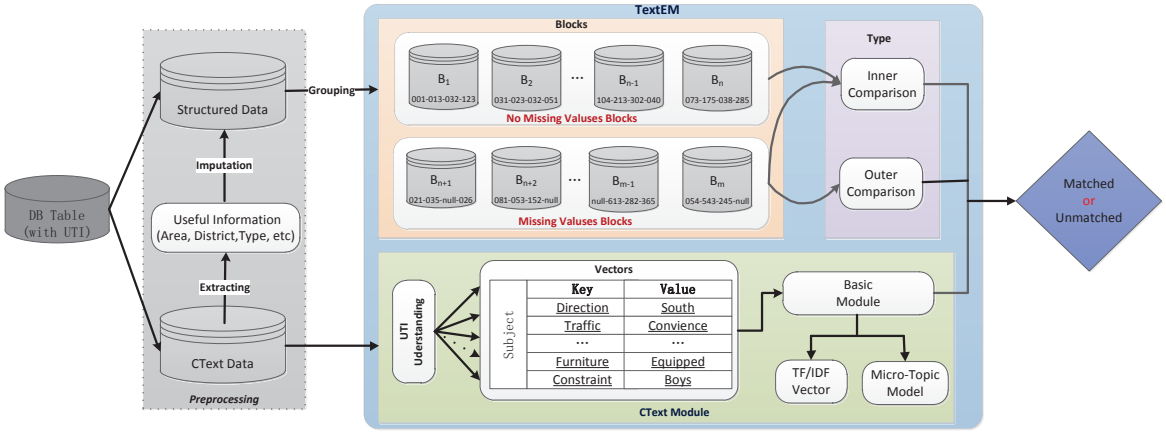


Fig. 1. Workflow Overview of CTextEM (Figure need to be updated! blocking, buckets, having missing values, tf/idf, the whole structure)

2 Problem Definition

Given a relational table, Entity Matching (EM) identifies all records referring to the same entity within the table. In this paper, we consider tables with both structured information (some might be missing) and CTexts. Particularly, we call the EM task employing CText as **CTextEM**.

The basic workflow of performing CTextEM is depicted in Fig. 1: We first rely on well-structured attribute values to group all records into different blocks, and then use the information in CText to do further EM within or across blocks. In the following, we briefly describe the key steps below:

1) Grouping Records into Blocks: We find a set of structured attributes A_s such that two records can not be matched if they do not have the same attribute values under A_s. We put those records sharing the same set of values under A_s into one block. However, a special case is that we may have records with missing values under A_s. We put those records having missing values under the same attributes and share the same values under the other attributes into one block.

Therefore, we actually have two kinds of blocks: blocks without missing values, and blocks with missing values. To perform EM, we not only need to do EM within every block, but also need to do EM across some block pair if the two blocks are possible to have matching records.

2) EM within Blocks: For records within either kind of block, we need to do EM between these records by employing the CText. That is, the records in the same records should be compared each other no matter what the kind of block is.

3) EM between Blocks: For those records in a block B_{n+1} with missing values, we also need to compare them with those records in some other blocks without missing values except for the difference of "null" that may possibly match with

these records in B_1. That is to say, the records without "null" and with "null" should also be compared to compute the similarities.

For performing EM either within block or between blocks, the key challenge lies on how we acquire useful information from CText for the EM task. More formally, we define the CTextEM problem as follows.

Definition 1. *Given a relational table* $T = \{r_1, r_2, ..., r_n\}$ *under the schema* $S = \{[A_1, A_2, ..., A_m], A_U\}$, *where* m, n *are positive integers,* r_i $(1 \leq i \leq n)$ *denotes a record,* A_j $(1 \leq j \leq m)$ *denotes an attribute with structured data, and* A_U *denotes the attribute with CText.* $\forall r_i, \forall r_j (1 \leq i, j \leq n, i \neq j)$ *in relation table* T, *CTextEM problem aims at finds a function* $\mathcal{F}(r_i, r_j, S)$ *and a threshold* θ, *if and only if:* $\mathcal{F}(r_i, r_j, S) \geq \theta$, *they are a pair of linked instances referring to the same entity. Otherwise, they are not matched instances.*

3 CTextEM Algorithms

In this section, we firstly present a baseline algorithm based on TF/IDF scores of phrases only, and then put forward an advanced topic-based algorithm that can identify information of different topics of the CText in a fine-grained way.

3.1 Baseline: Iterative TF/IDF-based CText Understanding

A baseline algorithm can be developed based on the TF/IDF scores of the phrases extracted from CText. Intuitively, we suppose that a set of phrases with the highest TF/IDF scores can represent the CText. Thus, our similarity function will be calculating the similarity between the two sets of phrases of the two CTexts.

1. Basic Workflow. Particularly, given a CText of the record, we consider all 1-5 word-length phrases from CText as candidate phrases after removing stop-words. Next, we calculate TF/IDF scores of these phrases and then select phrases to build up the comparison vectors. After that, we calculate the similarity between CTexts, and compare the result with a reasonable threshold. More details are given below:

a) *Calculating TF/IDF.* We first calculate the TF/IDF score of every phrase. Then we sort these phrases based on their TF/IDF score in ascend way. Particularly, the IDF score of a phrase is calculated based on the groups within the same blocks not based on the all records in the relation table. The reason why we do it like this is that the way is closer to fact that the similar things trend to owning the partial similarity not the overall similarity. In this way, we can decrease the cost of comparison among dissimilar records from different blocks.

b) *Building the Comparison Vectors.* Given an instances pair $(r_i, r_j)(1 \leq i, j \leq n, i \neq j)$, assume that r_i and r_j have x candidate phrases and y candidate phrases respectively, which can be showed with $v_i = \{w_1, w_2, ..., w_x\}$ and $v_j = \{w'_1, w'_2, ..., w'_y\}$ respectively. We compute the union-set $\bigcup(v_i, v_j)$ of

$v_i \cup v_j$ to get the common vector v_c whose size is $r(1 \leq r \leq (x+y))$. Next, we give the vector v_i, v_j the value set $\boldsymbol{v}_{vali}, \boldsymbol{v}_{valj}$ respectively. Uniformly, we normalize v_i and v_j with the method that if the current element of v_i exists in the v_c, the element of $\boldsymbol{v}_{vali}$ is "1", others "0". The process is also fit for the vector v_j. Assume that the size of v_c is z. Finally, we get the comparison vectors $\boldsymbol{v}_{vali} = \{bool(w_1, v_c), bool(w_2, v_c),$ $..., bool(w_z, v_c)\}$ and $\boldsymbol{v}_{valj} = \{bool(w'_1, v_c), bool(w'_2, v_c), ..., bool(w'_z, v_c)\}$ corresponding to r_i and r_j respectively, where $bool(\cdot)$ is a boolean function.

$$bool(w_i, v_c) = \begin{cases} 1, & \text{if } w_i \text{ exists in } v_c \\ 0, & \text{others} \end{cases} \tag{1}$$

c) *Computing the Similarity.* Given comparison vector $\boldsymbol{v}_{vali}$ and $\boldsymbol{v}_{valj}$ for r_i and r_j respectively, we compute the Cosine similarity between CText, which can be showed formally as follows:

$$sim(s_i, t_j) = \frac{\boldsymbol{v}_{vali} \times \boldsymbol{v}_{valj}}{\sqrt{v_{vali}^2 + v_{valj}^2}} = \frac{\sum_{m=1}^{r} bool(w_m, v_c) bool(w'_m, v_c)}{\sqrt{\sum_{p=1}^{r} bool(w_p, v_c)^2} \sqrt{\sum_{q=1}^{r} bool(w_q, v_c)^2}} \tag{2}$$

If $sim(s_i, t_j) > \theta$, the instance pair (s_i, t_j) will be linked and put into the same buckets, where θ is a user-defined similarity threshold.

2. Iterative Updating IDFs. As we process with the three steps above, the blocks of records are changed and so the IDF scores of phrases will be also different. The intuition of interaction is derived from the fact that: 1) as more matched entities are found, more relevant documents can be utilized for calculating the IDF score, 2) as more correlative CText are in the same blocks, we can find more matched entities. Thus we will iteratively update the IDF scores of all phrases and then repeat the above three steps, until the IDF scores become stable.

Briefly, the process of interaction is to update TF/IDF score and matched results constantly by running on the KPTI algorithm. For every record in the same block, we first compute its TF/IDF score based on new document library continually. Next, we build up new comparison vectors using the phrases of the previous steps. We finally compute the similarity of records and decide whether they are matched or not based on similarity threshold θ. Importantly, what is the stopping criteria of the process of interaction?

The process of interaction will go on continuously to find those matched entity pairs that should be linked but not due to low TF/IDF score of phrases. We will not stop the process of interaction until no more entities are found in the relation table. The interaction makes the precision and recall improved of EM to some extent.

3.2 A Micro-Topic-based CText Understanding Algorithm

The baseline algorithm measures the similarity between two CTexts in one dimension only. However, as a compound information of attribute, there are ac-

Algorithm 1: An Interaction Algorithm

Input : Two Relational Tables S and T with the Structured Attributes Set $S_1 = \{A_1, A_2, ..., A_l\}$ and the Structured Attributes Set $S_2 = \{A_{l+1}, A_{l+2}, ..., A_k\}$

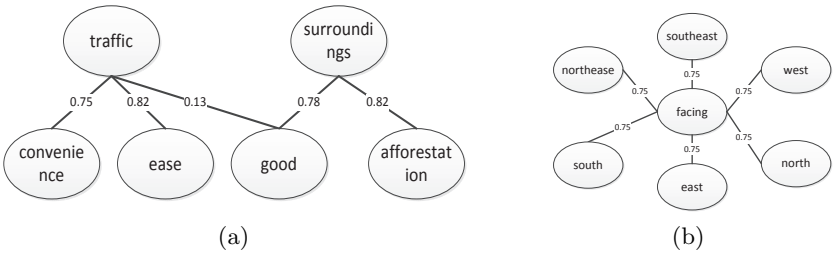
Output: Matching Entity Pairs

1. Set $\mathcal{O} = \mathcal{I} = \oslash$, $o = i = null$, where $\mathcal{O}, \mathcal{I}$ stand for the set of matching pairs and TF-IDF Document Libraries we have got separately, o, i stand for current pairs and TF-IDF document separately;
2. Initial Blocking based on Attributes Set S_1;
3. **for** $i = 0; i < n; i++$ **do**
    4. **for** $j = 0; j < n; j++$ **do**
        5. **if** r_i, r_j *are in the same big buckets, but not in small buckets* **then**
            6. **while** *more element* o *or* i *can be added into* $\mathcal{O}$ *and* $\mathcal{I}$ **do**
                7. Quadratic Blocking based on Attributes Set S_2 ;
                8. Build the Cosine vector for s_i, t_j to compute the similarity sim based on $sim(r_i, r_j)$;
                9. **if** $sim > \gamma$ **then**
                    10. Put r_i, r_j into the same small buckets;
                    11. Refresh $\mathcal{O}$;
                    12. Put the New CText into $\mathcal{I}$;
                    13. Refresh the document libraries $\mathcal{I}$;
                    14. Compute the new values of TF-IDF based on $\mathcal{I}$;
                end
            end
        end
    end
end
return $\mathcal{O}$;

tually information of different micro-topics in each CText. Different from topics such as "sports", "music" and "education" etc., the micro-topics can be taken as various aspects of the same topic. For instance, in the house renting information there are some aspects about direction, greening, property, traffic and so forth for describing the situation of house using CText.

In this subsection, we introduce a novel algorithm that works on mining micro-topics from CText, and then calculating the similarity between CTexts on all micro-topic dimensions. In the following, we firstly introduce how we build up the phrases relationship graph(PRG for short) employing CText by our training. Next, we introduce a greedy algorithm to pruning the PRG mentioned above to filter those unimportant nodes and edges. Last but not the least, we employ the phrase association degree(PAD for short) to measure the similarity of corresponding micro-topics for entity pairs, and translate the improved PRG into micro-topic model to build up the comparison vectors for calculating the similarity of entities.

Prof. Li stopped here, will continue later...



(a) (b)

Fig. 2. Example of PRGs

1. Constructing Phrases Relationship Graph. The phenomenon that some phrases occur with other phrases shows that there exist the association relationship among phrases. That is, some phrase always appear with some related phrases in some sentences at the same time, not others. Given a CText *ct*, we then employ the Longest-Cover [14] method to segment the subCText for getting the longest terms in the given vocabulary on the condition of filtering irrelevant stopping words. Next, we add edges among the results of word segmentation in order to show the PAD. To estimate the weight of edges, we compute the frequency of phrases appearing in the same CText. The weight between two phrases p_i and p_j can be calculated with the following formula by our training:

$$Fre_{ct}(p_i, p_j) = num_{ct} \cdot e^{-gap_{ct}(p_i,p_j)} \cdot bool(p_i, p_j) \tag{3}$$

where num_{ct} denotes the number of *ct* appearing in the training set, and $gap_{ct}(P_i, P_j)$ presents the distance of phrase p_i and p_j in the CText and $e^{-gap_{ct}(p_i,p_j)}$ is to penalize long distance of related phrases and $bool(p_i, p_j)$ is to reduce the influence of similar phrases in the same CText. The function $sim(\cdot, \cdot)$ computes the similarity of phrases, such as Edit-distance and *theta* is the string similarity threshold.

$$bool(p_i, p_j) = \begin{cases} 1, & \text{if } sim(p_i, p_j) \leq \theta \\ 0, & \text{otherwisw} \end{cases} \tag{4}$$

Next, we count up the total frequencies for all CText in training set, denoted by T, with $Fre(p_i, p_j)$, which can be calculated by:

$$Fre(p_i, p_j) = \sum_{ct \in T} Fre_{ct}(p_i, p_j) \tag{5}$$

Finally, we calculate the weight(namely PAD) of edge linking P_i to P_j with the following formulation:

$$PAD(p_i, p_j) = \frac{Fre(p_i, p_j)}{\sum\limits_{ct \in T} Fre_t(p_i, p_z)} \cdot \log \frac{Num}{Num_{adja(p_j, \bar{p}_i)}} \tag{6}$$

where $\frac{Fre(p_i,p_j)}{\sum_{ct \in T} Fre_t(p_i,p_z)}$ shows the probability p_j appearing with p_i at the same time, $log\frac{Num}{Num_{adja(p_j,\bar{p}_i)}}$ stands for penalizing phrases that co-occur with almost

other phrases. Besides, Num is the total number of nodes of the PRG and $Num_{adja(p_j,\bar{p}_i)}$ denotes the number of nodes adjacent to p_j except for p_i in the PRG.

Example 1. We get some PRGs by formulations (3)-(6) in our training set. Take two PRGs for example, as is shown in fig. 2, the left PRG shows one PRG links weakly to another PRG with one edge on the basis of instinct and the right one shows a single graph. The former reflects that the phrase "good" not only occur with phrase "traffic", but also appear with phrase "surroundings". However, the PDA is lower than other PDA obviously. The later shows that the phrase "facing" always occur with phrases localizers such as "south", "north", "southwest" and so forth.

2. Pruning the PRGs. As we can see in the fig 2, there are some weak association relationship between PRGs. These low PDA, however, have a bad effect on understanding the CText factually.

Our purpose is to delete the edges of PRG so that the correlations among a sub

and divide them into disjointed ones. Specifically, we measure the degree of clustering

translate our problem into the following optimization problem.

$$maximize \sum_{p_1 \in E_{p_1}, p_2 \in E_{p_2}} \frac{PAD(p_1, p_2)}{dis(p_1) + dis(p_2) + \alpha} \tag{7}$$

$$\begin{cases} dis(p_1) = Max_{p_1 \in E_{p_1}, p \in P} PAD(p_1, p) - Min_{p_1 \in E_{p_1}, p \in P} PAD(p_1, p) \\ dis(p_2) = Max_{p_2 \in E_{p_2}, p \in P} PAD(p_2, p) - Min_{p_2 \in E_{p_2}, p \in P} PAD(p_2, p) \end{cases} \tag{8}$$

where α is equilibrium factor to prevent the denominator being zero. And E'_p is the set of edges linked with p', P is the set of phrase domain.

Theorem 1. *Finding the optimal solution for objective 7 is NP-hard.*

Proof: We prove that the optimal solution is NP-hard even in the constraint of given the number of micro-topics. And we then prove it by reduction from the balanced maxskip partitioning problem [22]. Given a set V of binary vectors, where $|V|$ is a multiple of p, find a partitioning $\mathcal{P}$ over V such that the following total cost $\mathcal{C}(\mathcal{P})$ is maximized:

$$\mathcal{C}(\mathcal{P}) = \sum_{P_i \in \mathcal{P}} C(P_i) \tag{9}$$

where $C(P_i) = |P_i|$ is the cost of P_i and i is a constant. In our case, we denote the cost of a PRG P_i as

$$(P_i) = \sum_{(p_1,p_2) \in E} \frac{PAD(p_1, p_2)}{dis(p_1) + dis(p_2) + \alpha} = \sum_{(p_1,p_2) \in E} 1 - \Delta(P_i) \tag{10}$$

where $\Delta(P_i)$ is similar to $\bar{v}(P_i)j$ in the blanced maxskip partitioning problem. Thus, the objective 7 is equivalent to maximizing the total cost of $\mathcal{P}$, i.e. finding the optimal solution for objective 7 is NP-hard. Hence, theorem 1 is proved. □

As we all know, it is hard for solving a non-linear optimization problem. Therefore, we employ the approximating solution to optimize the question with greedy algorithm to select the important phrases and edges with high PAD of PGRs.

a) Node Selection.

- Rank the PAD of all edges in a descending way for every PRG.
- Rank the nodes of every PRG based on $Score(P_i)$ in a descending way, where

$$Score(P_i) = \frac{pad_1 + pad_2}{pad_1 - pad_2 + \alpha} \tag{11}$$

  pad_1 and pad_2 are the highest pad for node P_i in the PRG. Assume that $\mathcal{V}$ is the set of top k nodes for every PRG whose edge set is corresponding to E.
- For every node $p' \in \mathcal{V}$, select the edges with top two pad from E'_p, calculate the $PScore$ of nodes p' with the following formulation:

$$PScore(p') = \frac{\sum_{p \in E_{p'}} PAD(p', p)}{Max_{p \in E_{p'}} PAD(p', p) - Min_{p \in E_{p'}} PAD(p', p) + \alpha} \tag{12}$$

  Put more edges in the descending way of pad continually until PScore does not increase.

b) Edge Selection.

- Rank the nodes of every PRG based on $Score(P_i)$ in a descending way.
- Select edges greedily on the condition of increasing the score, until no such nodes.

Example 2. As we can see in fig. 2(a), the left PRG whose edge between "trafic" and "good" is pruned to be two separate PRGs with our pruning algorithm.

3. Acquiring the Topic. We acquire the micro-topics from the training set based on the results of step two. For every subCText of PRGs, we choose the edge with the highest PAD, regard linking nodes as candidate elements, compute the average PAD $averge(p)$ of candidate elements except for the highest PAD. The node with higher $averge(p)$ is the micro-topic. We build up the vector model with N micro-topics of PRGs in the form of $mod(mt_1, mt_2, ..., mt_N)$. For every dimension of mod, we employ domain knowledge to set the weight of different micro-topics for matching, whose identification degree is in the form of the set $weight(w_1, w_2, ..., w_N)$. The weight of every dimension for mod can be learned by learning model. We initially set $w_k = 1(1 \leq k \leq N)$.

We name the ability of attributes to identify one entity being different from another with the micro-topics as the probability p_{mt}, which can be calculated

from the training set. That is, on the one hand, how many p_{mt} is for entity r_i and r_j being the same one when the values are same on the dimension of micro-topic mt. And on the other hand, how many p_{mt} is for entity r_i and r_j being the same one when the values are different on the dimension of micro-topic mt. We iteratively calculate the weight of every micro-topic using the following formulation until it reaches to a stable value:

$$W(mt) = \frac{Pos_T(mt)}{Pos_T(mt) + Neg_T(mt)} \tag{13}$$

where $Pos_T(mt)$ is the number that if $r_i[mt] = r_j[mt]$, the entity pair (r_j, r_k) is linked together, while $Neg_T(mt)$ is the number that if $r_i[mt] \neq r_j[mt]$, the entity pair (r_j, r_k) is matched together.

Example 3. In fig. 2, from the left one the node "facing" has the highest $averge(\cdot)$, so it becomes one of micro-topic. Besides, the node "" is another micro-topic.

4. Matching Entities. Given entity pairs (r_i, r_j) in the same blocks from testing set, we should decide the subCTexts of r_i and r_j whether they belong to the same micro-topic. Two types of methods are provided for resolving this problem.

- If there both exist obvious micro-topic phrase for subCTexts of r_i and r_j, we calculate the similarity between two record with the adjusted cosine similarity [21]. Typically, for two records r_1 and r_2, we calculate their similarity as the following equation.

$$Sim(r_1, r_2) = \frac{\sum_{i=1}^{N} W^2(mt_i) \cdot sim(r_1[mt_i], r_2[mt_i])}{\sum_{j=1}^{N} [sim(r_1[mt_i], r_2[mt_i]) \cdot W(mt_i)]^2} \tag{14}$$

- If there exists one missing micro-topic phrase for ct for r, we then use the probabilistic model to deduce which topic it belongs to. Denoted $Pr(b|z)$ is ... b is ... z is ... . Notice that we can obtain $Pr(b|z)$ by some prior knowledge, thus we can get the following equation by the *Law of total probability*.

$$Pr(z|b) = \frac{Pr(b|z) \cdot Pr(z)}{\sum_z Pr(b|z) \cdot Pr(z)} \tag{15}$$

  ...
  After we obtain the topics of all the subCText, we employ the first case to build up comparison vector.

We employ the comparison vectors $vec(r_i)$ and $vec(r_j)$ to compute the similarity corresponding to entities r_i and r_j based on the formulation(2). If $sim(r_i, r_j) > \theta$, the instance pair (r_i, r_j) will be linked together, where θ is a user-defined similarity threshold.

Example 4. Give an example of matching results...

4 Experiments

We now present a thorough evaluation of our techniques. We compare our proposed methods with only Blocking algorithm, Key based method and Decision Tree method. Our first database is collected from three house renting information websites, *ganji, anjuke, 58tongcheng* of ten large-medium cities of China: *Beijing, Shanghai, Guangzhou, Shenzheng, Wuhan, Nanjing, Tianjing, Hangzhou, Chendu, Suzhou*. In our experiments, we regard city as a unity to do EM with above methods. That is, every data set is from three mentioned above websites of a city. The property of house renting information can be seen in the Table 2. The second database is crawled from ganji website and "The home of used-car" website, which contain the information of second-cars including structured data and CText information. The property of used-car data can be shown in the Table 3.

	city				
	beijing	chengdu	suzhou	shenzhen	tianjin
Attribute Number	22	22	22	22	22
Record Number	253750	324990	397630	433710	371250

Table 2. The house renting information from five cities

	website	
	ganji	The home of used-car
Attribute Number	22	22
Record Number	433710	371250

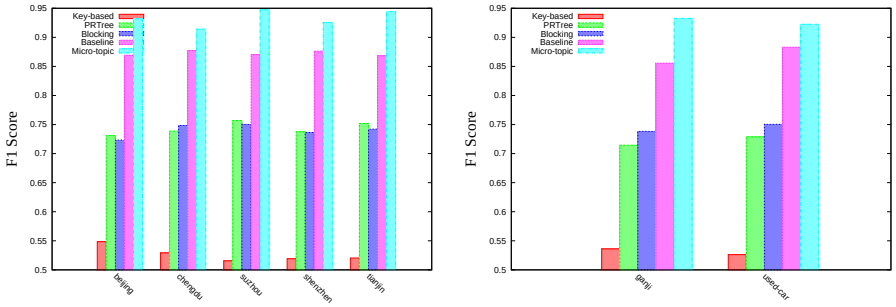
Table 3. The used-car information from two websites

We basically use three metrics to evaluate the effectiveness of the methods: **Precision:** the percentage of correctly linked instance pairs among all linked instance pairs, **Recall:** the percentage of correctly linked instance pairs among all instance pairs that should be linked, and $F1$ **Score:** a combination of precision and recall, which is calculated by $F1 = \frac{2*precision*recall}{precision+recall}$. We use the time cost of an algorithm for evaluating the efficiency of a method.

4.1 Comparison with Previous Methods on F1

In this section, we compare our methods *Key Phrases Based on Baseline, Microtopic Based* with traditional methods *Key Based, Blocking Based, the Probabilistic Rule-based Decision Tree (PRTree for short).*

The method *Key Based EM* combines many kinds of state-of-art methods based on key values from [1, 6, 24], such as Q-gram, prefix-based filtering and Inverted indices. This way can decrease the comparison cost greatly to improve the efficiency. It is , however, vulnerable to the different representation of same entities. And *Blocking Based EM* aims to select some attributes with high identification to create hash buckets for matching entities. The entities in the



(a) The House Renting Information of Five Cities

(b) The Used-car Information of Two Websites

Fig. 3. The Comparison with Previous Methods on F1

same buckets are likely to be the same, while the entities with different hash codes can not be same [4]. Besides,*PRTree* [28] build up a probability rule-based decision tree utilizing two characteristic(*Sufficiency and Necessary*) of attribute to get the matching model first, then iterates from the root node to leaf nodes of PRTree to do EM, where *Sufficiency* of attributes denotes the ability to assert entity pairs are same, while *Necessary* of attributes represents the ability to ensure entity pairs are not same.

As shown in Fig 3(a), we can see that the Micro-topic EM better than all kinds of EM algorithms. This is consistent with our expectations. The Key Based EM with the lowest effectiveness because of the different forms of key values. The effect of Blocking Based EM are discounted greatly due to the missing values of structured data, which leads to the occurrences of *the false positive*. The PRTree EM is better than the Key Based EM, while it is worse than the Baseline. Since PRTree choose the attributes with highest Sufficiency score or Necessary score as the node of tree, it will find the matched entities, while it shows bad owing to insufficient information to apply and the low score of Sufficiency and Necessary. The Baseline algorithm extracts information from CText combining structured data to do EM, which results in good effectiveness. The micro-topic mines the relationship of phrases in a fine-grained way using undirected graph. Therefore, it can get more accurate data to build up the comparison vectors. The case above is suitable for the Fig 3(b).

4.2 Precision and Recall of Methods

In the following, we compare these methods in Precision and Recall on five cities. As can be observed in Table 4, the Micro-topic EM gets the highest effectiveness, the Baseline following it. While they employ the CText information to acquire the key phrases for EM, the key phrases of Baseline is too rough to get the accurate information. The PRTree EM and Blocking EM are neck and neck. The Key Based EM has the lowest effect. Because insufficient information can be used for assist EM leading to non-matched.

4.3 The Result Extracted from CText: Baseline vs. Micro-topic

We now show the key information extracted from CText with Baseline and Micro-topic. From Table 5, we can see that the performance of Micro-topic EM is better than Baseline EM, since it can filter some irrelative phrases which are included in the results of Baseline. One the one hand, the former improves the EM efficiency in that these irrelative phrases are rejected to participating in

Methods	beijing		chengdu		suzhou		shenzhen		tianjin	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Key Based	0.6994	0.4512	0.7116	0.4212	0.7254	0.3998	0.7059	0.4105	0.7142	0.4093
PRTree	0.7504	0.7125	0.7542	0.7239	0.7556	0.7582	0.7694	0.7081	0.7562	0.7472
Blocking	0.7452	0.7028	0.7645	0.7332	0.7583	0.7425	0.7467	0.7259	0.7556	0.7293
Baseline	0.8966	0.8437	0.9059	0.8498	0.8891	0.8524	0.9105	0.8447	0.8725	0.8639
Micro-topic	0.9688	0.8974	0.9472	0.8836	0.9802	0.9163	0.9650	0.8892	0.9823	0.9089

Table 4. Comparison of Methods in Precision and Recall for Five Cities

Methods	Example	
	1. Community Planning well and unique warmth, flowers and trees patchwork, like a garden, furniture and appliances equipped well. 2. Hardcover house, well-groomed room very much, matching color, facing south right, twenty floor.	1. south facing, good lighting, two air conditioning, water heaters and washing machines are proved, free of property charges
Baseline	Community Planning, warmth, flowers, trees, garden, furniture, appliances, Hardcover, house, well-groomed, very much, south,twenty, floor	south, lighting, two, air conditioning, water heaters, washing machines, free of, property charges
Micro-topic	Community Planning, warmth, flowers and trees, furniture and appliances, Hardcover, well-groomed,south, floor	south, lighting, air conditioning, water heaters, washing machines, property charges

Table 5. The comparison of Extracted Information of Baseline and Micro-topic

calculating the similarity of entities. On the other hand, the EM effectiveness is also enhanced since more accurate phrases are utilized to do EM.

4.4 The Weight of Key Phrases for Micro-topic

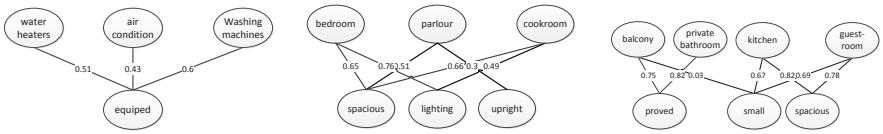
As illustrated in Table 6, we can see that different key phrases in the PRG have different weights. It is consistent with our exceptions that the phrases with high discrimination owns bigger weight than others. Since they always co-occur with some phrases to enhance the relationship among them. For example, the phrase "facing" appears with "south", "north", "northwest" and so forth.

CText	1. Community Planning well and unique warmth, flowers and trees patchwork, like a garden, furniture and appliances equipped well. 2. Hardcover house, well-groomed room very much, matching color, facing south right, twenty floor.							
Phrases	Community Planning	warmth	flowers and trees patchwork	furniture and appliances	Hardcover	color	south	floor
Weight	0.2	0.34	0.32	0.4	0.69	0.5	0.85	0.89

Table 6. The used-car information from two websites

4.5 Different Types of PRGs

We get the different types of PRGs on the training set, since some micro-topics are independent with others, and others are interval for each other. Besides, some micro-topics have the weak relationship with others such that this kind of relationship can be ignored. As is shown in Fig 4(a), the micro-topic "equiped" is independent with other micro-topics. In Fig 4(b), we can see that the micro-topic "spacious", "lighting" and "upright" are interval for each other owning the strong relationship. The micro-topic "proved" has the weak association with the micro-topics "small" and "spacious".

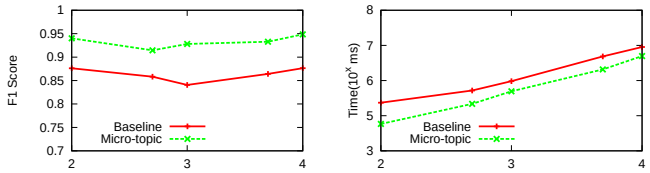


(a) Single PRG (b) Interlaced PRGs (c) The Weak Relationship between PRGs

Fig. 4. The Comparison with Previous Methods on F1

4.6 Scalability

We also evaluate the scalability of Baseline and Micro-topic. As illustrated in Fig 5(a), with the records number changing from 100 to 10000, the F1 Score of Micro-topic EM trends to growing on the whole. Besides, Micro-topic EM is more stable than Baseline Algorithm. The former starts to rise when getting the second point, while the later stills to decline. In Fig 5(b), we can see that the time cost of Micro-topic EM is less than Baseline Algorithm obviously. And the trend of growth of time cost for Micro-topic EM is more tempolabile than Baseline.



(a) #Records(10^x) (b) #Records(10^x)

Fig. 5. The Efficiency of Baseline and Mirco-topic in Different Number of Records

5 Related Work

So far, plenty of work has been done on EM based on the string similarities [15], correlations [20], or semantic similarity [7] between various kinds of structured attribute values of the records such as digital values, date values or short string values in EM (see [10] for a survey). However, EM based on structured information only may easily fail when the structured information is not enough to identify the matching relationships between records.

As a complementary to structured information, we often have some unstructured textual information with each record, which we call as CText for short. Since there can be dozens of sentences (or thousands of words) with each CText, the conventional string similarity metrics such as edit-distance [27], jaccard [8], bag-of-words [19] and some hybrid metrics [5, 26] can not be applied.

To utilize the information in CText for EM, the key is to identify useful information from noises, a big challenge is how to identify the key information and eliminate the useless words [25]. Recently, some work has been done for unstructured information. An approximate data matching method [23] is proposed, which employs semantic information to compare whether the candidate entities are matched together. But this method is based on special domain knowledge and limited to semi-structured information. A model based on unstructured text are present in [12], which arrives at a good precision and recall demonstrated with DBWorld posts. However, it needs the support of a special ontology largely. What is more, Ektefa et al. [8] proposes a threshold similarity measure for EM, which employs a combination method of string similarity and semantic similarity measures. But the measure is not robust in that the string values in experiments are immune to WordNet [17] due to lacking the normal English terms.

Besides, there are some topic models algorithms to discover the main themes for text information in the filed of *Nature Language Processing(NLP)*, such as *Latent Dirichlet Allocation(LDA)* [3], *Latent Semantic Analysis(LSA)* [16] and *Probabilistic Latent Semantic Analysis(PLSA)* [13]. They can get the hidden variables named topic words from text with some machine learning algorithms. However, these methods will fail without the obvious topic of text to get the useful information from CText.

There are also some researches on Text Understanding and Text Summarization which is similar to our work. Some kinds of methods on Text Summarization ([9] for a survey) are proposed for understanding the meaning of CText. Amini et al. [2] proposes a new approach for Single Document Summarization based on a Machine Learning ranking algorithm for text summarization. Neto et al. [18] employs trainable Machine Learning algorithms based on a set of features extracted directly from the original text for addressing the automatic summarization task. [] apply deep learning to text understanding from character level inputs all the way up to abstract text concepts, using temporal convolutional networks. They are all devoted to learning about the main idea of CText rather than considering the relationship among phrases from CText.

6 Conclusions and Future Work

Existing EM methods seldom consider to utilize the unstructured information in EM due to the difficulty in extracting the key information. We propose two novel approaches that can deal with unstructured text information combination with traditional structured data for EM. We introduce one type of effective blocking algorithm to decrease the cost of comparison among records in the relation tables. Our approaches for CText can extract key information employed in constructing vectors for computing similarity among records. The Micro-topic EM aims to get the PRG of key phrases to build up the comparison model, which is gain the key phrases(information) which are not useful for EM possibly in coarse-grained way, while our method *Key Phrases based on Feature* can obtain feature

phrases of CText in a fine-grained manner to compute similarity. Both of them can enhance the effectiveness of EM compared to traditional methods.

Besides, we propose an interaction algorithm between EM and *Key Phrases based on TF-IDF* to get more matched entities. Note that our methods is not immune to missing values for EM. Extensive experimental results based on several data collections demonstrate that our proposed *Key Phrases based on Feature* algorithm can improve the efficiency of EM on average ?% cost of the *Key Phrases based on TF-IDF* algorithm, and enhance the efficiency of EM at some extent than traditional methods. Future work may consider combining Crowdsouring with our methods to get more accuracy phrases for EM, which can improve effectiveness further more.

Acknowledgment

The authors would like to thank...

References

1. A. Aizawa and K. Oyama. A fast linkage detection scheme for multi-source information integration. In *Web Information Retrieval and Integration, 2005. WIRI '05. Proceedings. International Workshop on Challenges in*, pages 30–39, 2005.
2. M. R. Amini, N. Usunier, and P. Gallinari. *Automatic Text Summarization Based on Word-Clusters and Ranking Algorithms*. Springer Berlin Heidelberg, 2005.
3. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003.
4. A. Borthwick, A. Goldberg, P. Cheung, and A. Winkel. Batch automated blocking and record matching, 2011.
5. D. G. Brizan and A. U. Tansel. A. survey of entity resolution and record linkage methodologies. *Communications of the IIMA*, 6(3):5, 2015.
6. P. Christen. A survey of indexing techniques for scalable record linkage and deduplication. *Knowledge & Data Engineering IEEE Transactions on*, 24(9):1537–1555, 2012.
7. R. Dhamankar, Y. Lee, A. Doan, A. Halevy, and P. Domingos. imap: discovering complex semantic matches between database schemas. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 383–394. ACM, 2004.
8. M. Ektefa, F. Sidi, H. Ibrahim, M. A. Jabar, S. Memar, and A. Ramli. A threshold-based similarity measure for duplicate detection. In *Open Systems (ICOS), 2011 IEEE Conference on*, pages 37–41. IEEE, 2011.
9. S. Elfayoumy and J. Thoppil. A survey of unstructured text summarization techniques. 5(4), 2014.
10. A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 19(1):1–16, 2007.
11. C. Gao, X. Hong, Z. Peng, and H. Chen. Web trace duplication detection based on context. In *Web Information Systems and Mining*, pages 292–301. Springer, 2011.

12. J. Hassell, B. Aleman-Meza, and I. B. Arpinar. *Ontology-driven automatic entity disambiguation in unstructured text.* Springer, 2006.
13. T. Hofmann. Probabilistic latent semantic analysis. *Proc of Uncertainty in Artificial Intelligence Uai*, 25(4):289–296, 1999.
14. D. Kim, H. Wang, and A. Oh. Context-dependent conceptualization. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2654–2661. AAAI Press, 2013.
15. N. Koudas, S. Sarawagi, and D. Srivastava. Record linkage: similarity measures and algorithms. *Sigmod Conference*, pages 802–803, 2006.
16. T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis [special issue. *Discourse Processes*, 25(2):259–284, 1998.
17. G. A. Miller. Wordnet: a lexical database for english. *Communications of the Acm*, 38(11):39–41, 1995.
18. J. L. Neto, A. A. Freitas, and C. A. A. Kaestner. *Automatic Text Summarization Using a Machine Learning Approach.* Springer Berlin Heidelberg, 2002.
19. H. T. Nguyen, C. Barat, and C. Ducottet. Approximate image matching using strings of bag-of-visual words representation. In *International Conference on Computer Vision Theory and Applications*, pages 345–353, 2014.
20. E. Parkhomenko, D. Tritchler, and J. Beyene. Sparse canonical correlation analysis with application to genomic data integration. (1):1–34, 2009.
21. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.
22. L. Sun, M. J. Franklin, S. Krishnan, and R. S. Xin. Fine-grained partitioning for aggressive data skipping. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1115–1126. ACM, 2014.
23. M. Szczuka, P. Betliński, and K. Herba. Named entity matching in publication databases. In *Rough Sets and Current Trends in Computing*, pages 172–179. Springer, 2012.
24. J. Wang, G. Li, and J. Feng. Can we beat the prefix filtering?: an adaptive framework for similarity join and search. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 2012.
25. S. M. Weiss, N. Indurkhya, T. Zhang, and F. Damerau. *Text mining: predictive methods for analyzing unstructured information.* Springer Science & Business Media, 2010.
26. W. E. Winkler. Overview of record linkage and current research directions. In *Bureau of the Census.* Citeseer, 2006.
27. C. Xiao, W. Wang, and X. Lin. Ed-join: an efficient algorithm for similarity joins with edit distance constraints. *Proceedings of the VLDB Endowment*, 1(1):933–944, 2008.
28. Q. Yang, Z. Li, J. Jiang, P. Zhao, G. Liu, A. Liu, and J. Zhu. Nokearm: Employing non-key attributes in record matching. *Lecture Notes in Computer Science*, 2015.