Li ZX, Yang Q, Liu A *et al.* Crowd-guided entity matching with consolidated textual data. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 32(5): 858–876 Sept. 2017. DOI 10.1007/s11390-017-1769-0

Crowd-Guided Entity Matching with Consolidated Textual Data

Zhi-Xu Li[1,2], *Member, CCF*, Qiang Yang[1], An Liu[1,*], *Member, CCF*, Guan-Feng Liu[1], *Member, CCF*
Jia Zhu[3], *Member, CCF*, Jia-Jie Xu[1], *Member, CCF*, Kai Zheng[1,4], *Member, CCF*
and Min Zhang[1], *Member, CCF*

[1] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*

[2] *Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou 510006, China*

[3] *School of Computer, South China Normal University, Guangzhou 510631, China*

[4] *Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing 100872, China*

E-mail: zhixuli@suda.edu.cn; qiangyanghm@hotmail.com; {anliu, gfliu}@suda.edu.cn; jzhu@m.scnu.edu.cn
{xujj, zhengkai, minzhang}@suda.edu.cn

Received March 1, 2017; revised August 9, 2017.

Abstract Entity matching (EM) identifies records referring to the same entity within or across databases. Existing methods using 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 databases may have some unstructured textual attribute containing extra consolidated textual information (CText) of the record, but seldom work has been done on using the CText for EM. Conventional string similarity metrics such as edit distance or bag-of-words are unsuitable for measuring the similarities between CText since there are hundreds or thousands of words with each piece of CText, while existing topic models either cannot work well since there are no obvious gaps between topics in CText. In this paper, we propose a novel cooccurrence-based topic model to identify various sub-topics from each piece of CText, and then measure the similarity between CText on the multiple sub-topic dimensions. To avoid ignoring some hidden important sub-topics, we let the crowd help us decide weights of different sub-topics in doing EM. Our empirical study on two real-world datasets based on Amzon Mechanical Turk Crowdsourcing Platform shows that our method outperforms the state-of-the-art EM methods and Text Understanding models.

Keywords entity matching, consolidated textual data, crowdsourcing

1 Introduction

With the data explosion for decades, the inconsistency between records becomes more and more serious within or 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[1] or correlations[2] among various kinds of structured attribute values such as digital, date or short string values (see [3] for a survey). However, methods based on structured information may easily fail for lacking enough information for EM.

Nowadays there are usually some long free-text description about records, such as those second-hand goods (like cars, houses, or furniture) for selling online (see Fig.1 for example), which have limited structured information but with a "general supplemental description" attribute containing some extra information like "orientation", "virescence", "type of decoration". Given that such a long free-text description contains

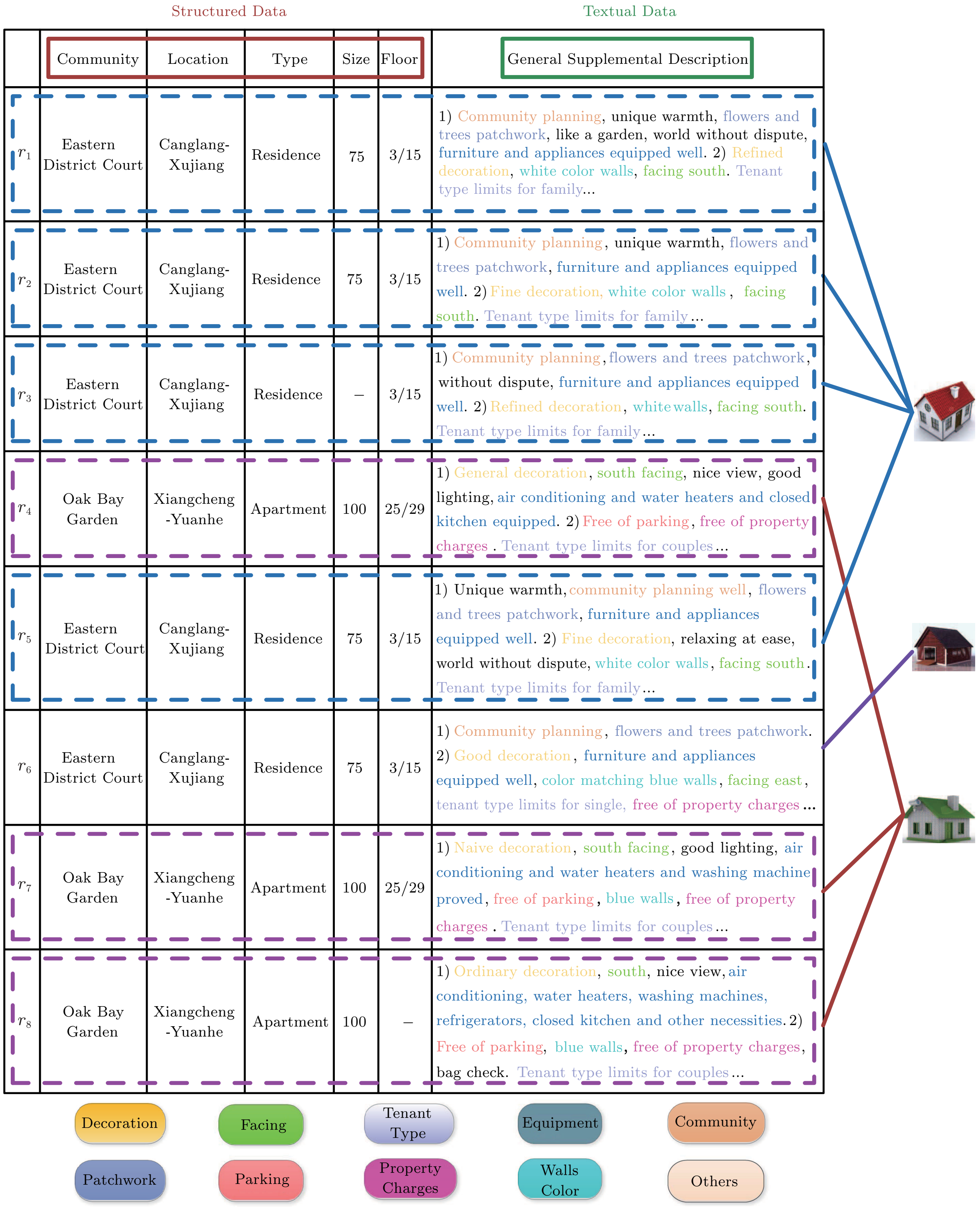
Regular Paper

Special Section on Crowdsourced Data Management

This research is partially supported by the National Natural Science Foundation of China under Grant Nos. 61632016, 61402313, 61303019, 61472263, and 61572336, the Postdoctoral Scientific Research Funding of Jiangsu Province of China under Grant No. 1501090B, the National Postdoctoral Funding of China under Grant Nos. 2015M581859 and 2016T90493, and the Open Foundation of Guangdong Key Laboratory of Big Data Analysis and Processing of China under Grant No. 2017012.

A preliminary version of the paper was published in the Proceedings of DASFAA 2016.

[*] Corresponding Author



	Community	Location	Type	Size	Floor	General Supplemental Description
r_1	Eastern District Court	Canglang-Xujiang	Residence	75	3/15	1) Community planning, unique warmth, flowers and trees patchwork, like a garden, world without dispute, furniture and appliances equipped well. 2) Refined decoration, white color walls, facing south. Tenant type limits for family...
r_2	Eastern District Court	Canglang-Xujiang	Residence	75	3/15	1) Community planning, unique warmth, flowers and trees patchwork, furniture and appliances equipped well. 2) Fine decoration, white color walls, facing south. Tenant type limits for family...
r_3	Eastern District Court	Canglang-Xujiang	Residence	–	3/15	1) Community planning, flowers and trees patchwork, without dispute, furniture and appliances equipped well. 2) Refined decoration, white walls, facing south. Tenant type limits for family...
r_4	Oak Bay Garden	Xiangcheng-Yuanhe	Apartment	100	25/29	1) General decoration, south facing, nice view, good lighting, air conditioning and water heaters and closed kitchen equipped. 2) Free of parking, free of property charges . Tenant type limits for couples...
r_5	Eastern District Court	Canglang-Xujiang	Residence	75	3/15	1) Unique warmth, community planning well, flowers and trees patchwork, furniture and appliances equipped well. 2) Fine decoration, relaxing at ease, world without dispute, white color walls, facing south. Tenant type limits for family...
r_6	Eastern District Court	Canglang-Xujiang	Residence	75	3/15	1) Community planning, flowers and trees patchwork. 2) Good decoration, furniture and appliances equipped well, color matching blue walls, facing east, tenant type limits for single, free of property charges...
r_7	Oak Bay Garden	Xiangcheng-Yuanhe	Apartment	100	25/29	1) Naive decoration, south facing, good lighting, air conditioning and water heaters and washing machine proved, free of parking, blue walls, free of property charges . Tenant type limits for couples...
r_8	Oak Bay Garden	Xiangcheng-Yuanhe	Apartment	100	–	1) Ordinary decoration, south, nice view, air conditioning, water heaters, washing machines, refrigerators, closed kitchen and other necessities. 2) Free of parking, blue walls, free of property charges, bag check. Tenant type limits for couples...

Fig.1. Example house renting information table with CText. Entities r_1, r_2, r_3 and r_5 are same while r_4, r_7 and r_8 are same.

various kinds of information on several sub-topics, we call it consolidated textual information (CText) in this paper. Therefore, why do we not use the CText for better EM?

Apparently, conventional string similarity metrics such as edit distance or bag-of-words are unsuitable for measuring the similarities between CText since there are usually hundreds or even thousands of words with each piece of 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.*[4] calculated both a string similarity score and a semantic similarity score between CText. 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 datasets. Gao *et al.*[5] put forward a semantic features based method, which defines a semantic feature vector like {*time, location, agentive, objective, activity*} for every piece of CText, and then trains a classifier to identify duplicate records based on their feature vectors. However, this method is also limited in the dimensions of the employed features, and thus cannot be easily applied to the other datasets.

Essentially, our problem is also very similar to Text Understanding[6], which focuses on understanding the information contained in unstructured text. Classical topic models such as Latent Dirichlet Allocation (LDA)[7], Latent Semantic Analysis (LSA)[8] and Probability Latent Semantic Analysis (PLSA)[9] could identify topics from free texts such as the topics of news about "education", "financial", "sports", "music" and so on. However, as a general description/metadata about a record, the topics in a piece of CText can be seen as sub-topics of a general topic, and thus they may share many topic words and there is no clear gap between these sub-topics such as the example shown in Fig.1. On the other hand, a sub-topic in a piece of CText can be very short (like several words). Thus we can hardly learn any sub-topic words with previous topic models.

Given the above, we would like to propose a novel algorithm that works on mining sub-topics from CText. Intuitively, if two phrases are always mentioned in the same sentences, it is quite possible that there exists some association relationship between the two phrases. Based on this intuition, we will build up a phrase coocurrence graph to denote the cooccurrence relationships among all phrases in CText. By doing proper partition on the graph, we expect to divide the graph into partitions, each of which corresponds to a sub-topic of the CText. We finally measure the similarity between two records on all sub-topic dimensions.

The key challenge here lies on how we perform graph partition to make each partition closely corresponding to a sub-topic. We model the problem into an optimization problem and then analyze in theory that this optimization problem is an NP-hard one. To solve the problem, a baseline method employs a so-called phrase association degree to measure the similarity between two records on corresponding sub-topic dimensions, and then propose a greedy algorithm that always selects the edge with the minimum phrase association degree as the point of partition. However, this method may have difficulties in estimating the importance (or weight) of different sub-topics in EM. In other words, it may underestimate some hidden important sub-topics and thus decrease the recall of EM. For example, the weight of the sub-topic "tenant type" might be very low for the "House Renting Information" dataset, but it is actually an important factor that we should pay attention to in doing EM. Given this, we propose to use the crowd as our guidance to help decide the weights of different sub-topics in doing EM. For this purpose, we work on estimating the accuracy of each worker and selecting the most suitable worker to fulfill generated tasks.

We summarize our contributions below.

• We work on a novel EM problem, using CText, and put forward a cooccurrence-based sub-topic analytics model that is able to acquire information on multiple sub-topics from the CText for more accurate EM.

• We model the key graph partition problem in our proposed EM method into an optimization problem. After analyzing that this problem is an NP-hard problem, we propose a greedy algorithm that always selects the edge with the minimum phrase association degree as the point of partition.

• We propose to use the crowd to further improve the performance of the proposed EM method using CText by allowing the crowd to help decide the weights of different sub-topics in doing EM. We find proper ways to estimate the accuracy of each worker and select the most suitable worker to fulfill every generated task.

The rest of the paper is organized as follows. We define the problem of EM using CText and give our workflow overview in Section 2, and then present our algorithms on using CText for EM in Section 3. After introducing how we use the crowd to help decide the weights of different sub-topics in Section 4, we report our experimental study in Section 5. Related work is covered in Section 6, and the paper is concluded in Section 7.

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 a set of structured attributes (some might be missing) and an unstructured attribute with CText. Particularly, we call the EM task employing CText as CTextEM. More formally, we define the CTextEM problem as follows.

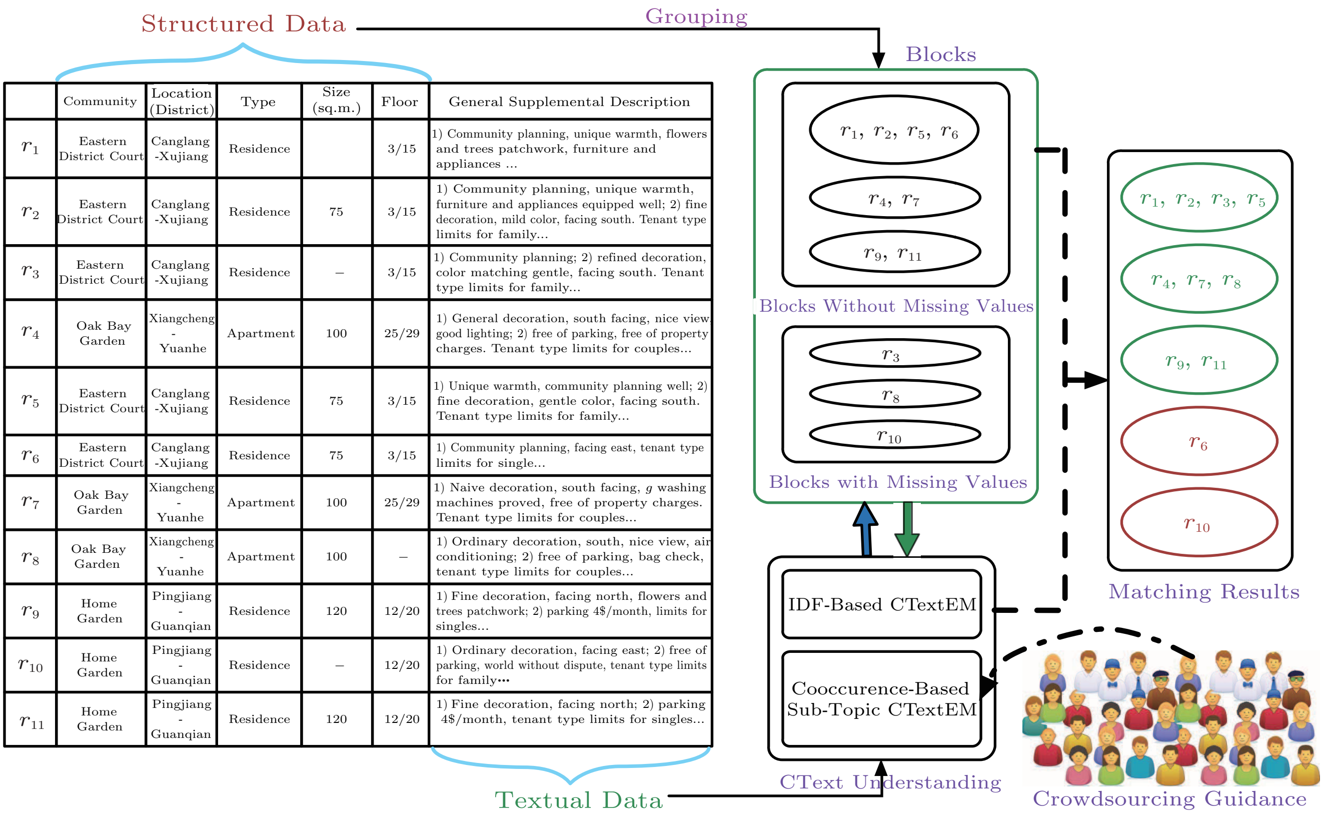
Definition 1 (CTextEM). *Given a relational table* $T = \{r_1, r_2, ..., r_n\}$ *under the schema* $S = \{(A_1, A_2, ..., A_m), A_U\}$, *where* m, n *are positive integers,* r_i *denotes a record* $(1 \leqslant i \leqslant n)$, A_j *denotes an attribute with structured data* $(1 \leqslant j \leqslant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

In this paper, we employ both structured attributes and CText for EM. The basic workflow can be depicted in Fig.2. We first rely on the structured attribute values to group all records into different blocks, and then use the information in CText to do further EM within or between blocks.

1) *Grouping Records into Blocks.* We find a set of structured attributes A_s which satisfy that: two records cannot be matched if they do not have the same attribute values under A_s. We put those records sharing the same A_s values into one block. A special case here is the records with missing values under A_s. We put those records having missing values under the same attributes in A_s while sharing the same values under the other attributes in A_s into one block.

2) *EM Within or Between Blocks.* For records within one block, we perform EM between every pair of records by employing the CText. Also, we say the records between two blocks B_1 and B_2 should also be compared pair-wisely, if the two blocks share the same codes under all non-"null" attributes.

For performing EM either within blocks or between blocks, the key challenge lies in how we acquire useful information from CText for the EM task. In Section 3,



	Community	Location (District)	Type	Size (sq.m.)	Floor	General Supplemental Description
r_1	Eastern District Court	Canglang -Xujiang	Residence		3/15	1) Community planning, unique warmth, flowers and trees patchwork, furniture and appliances ...
r_2	Eastern District Court	Canglang -Xujiang	Residence	75	3/15	1) Community planning, unique warmth, furniture and appliances equipped well; 2) fine decoration, mild color, facing south. Tenant type limits for family...
r_3	Eastern District Court	Canglang -Xujiang	Residence	–	3/15	1) Community planning; 2) refined decoration, color matching gentle, facing south. Tenant type limits for family...
r_4	Oak Bay Garden	Xiangcheng - Yuanhe	Apartment	100	25/29	1) General decoration, south facing, nice view good lighting; 2) free of parking, free of property charges. Tenant type limits for couples...
r_5	Eastern District Court	Canglang -Xujiang	Residence	75	3/15	1) Unique warmth, community planning well; 2) fine decoration, gentle color, facing south. Tenant type limits for family...
r_6	Eastern District Court	Canglang -Xujiang	Residence	75	3/15	1) Community planning, facing east, tenant type limits for single...
r_7	Oak Bay Garden	Xiangcheng - Yuanhe	Apartment	100	25/29	1) Naive decoration, south facing, *g* washing machines proved, free of property charges. Tenant type limits for couples...
r_8	Oak Bay Garden	Xiangcheng - Yuanhe	Apartment	100	–	1) Ordinary decoration, south, nice view, air conditioning; 2) free of parking, bag check, tenant type limits for couples...
r_9	Home Garden	Pingjiang - Guanqian	Residence	120	12/20	1) Fine decoration, facing north, flowers and trees patchwork; 2) parking 4$/month, limits for singles...
r_{10}	Home Garden	Pingjiang - Guanqian	Residence	–	12/20	1) Ordinary decoration, facing east; 2) free of parking, world without dispute, tenant type limits for family•••
r_{11}	Home Garden	Pingjiang - Guanqian	Residence	120	12/20	1) Fine decoration, facing north; 2) parking 4$/month, tenant type limits for singles...

Fig.2. Workflow overview of crowd guidance CTextEM.

we will mainly focus on introducing how we mine the information in CText for EM between records.

3 Using CText for EM

We first present a baseline algorithm based on inverse document frequency (IDF) scores of phrases, and then put forward a cooccurrence-based sub-topic analytics model for detecting sub-topics from CText.

3.1 Baseline: Iterative IDF-Based Method

A baseline algorithm supposes that a set of phrases with the highest IDF scores in a piece of CText can approximately represent the CText. Thus, our similarity function for calculating the similarity between two pieces of CText will be calculating the similarity between the phrase sets of the two pieces of CText.

1) *Basic Workflow.* Particularly, given a piece of CText of a record, we consider all 2∼6 word-length phrases from the CText as candidate phrases after removing stop-words. We calculate IDF scores of these phrases and then select phrases to build up the comparison vectors. After that, we calculate the similarity between CText, and compare the result with a predefined threshold. More details are given below.

a) *Building the Comparison Vectors.* We first calculate the IDF score of each phrase. Note that the IDF score of a phrase is calculated within each block. Second we sort these phrases based on their IDF scores in an ascend way and only use top-ranked phrases to represent a database record. Finally, we collect all different phrases from all records into a global phrase set $\boldsymbol{P}_g = (w_1, w_2, ..., w_g)$, according to which we can build a Boolean vector $\boldsymbol{v}_i = (bool(r_i, w_1), bool(r_i, w_2), ..., bool(r_i, w_n))$ for each record $r_i (1 \leqslant i \leqslant n)$, where

$$bool(r_i, w_j) = \begin{cases} 1, & \text{if } w_j \in r_i, \\ 0, & \text{otherwise.} \end{cases}$$

b) *Computing the Similarity.* Given the comparison vectors $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ for r_i and r_j $(1 \leqslant i, j \leqslant n)$ respectively, we compute the cosine similarity between the two pieces of CText as follows:

$$\begin{aligned} & sim(r_i, r_j) \\ = & \frac{\boldsymbol{v}_i \cdot \boldsymbol{v}_j}{||\boldsymbol{v}_i|| \times ||\boldsymbol{v}_j||} \\ = & \frac{\sum_{p=1}^{g} bool(r_i, w_p) \times bool(r_j, w_p)}{\sqrt{\sum_{p=1}^{g} bool(v_i, w_p)^2} \times \sqrt{\sum_{q=1}^{g} bool(v_j, w_q)^2}}, \quad (1) \end{aligned}$$

where $|| \cdot ||$ represents the norm of a vector.

c) *Adjusting Blocks.* Let θ denote the predefined similarity threshold. If $sim(r_i, r_j) > \theta$ and the two instances r_i and r_j are used to be in the same block, they will be merged into one record in the block where $sim(r_i, r_j)$ can be calculated with (1). Otherwise, if $sim(r_i, r_j) > \theta$ but the two instances r_i and r_j are used to be in different blocks, we will move r_j from the original block to r_i's block, and merge it with r_i, assuming that r_j's block is the block with missing values.

2) *Iterative Updating IDFs.* The intuition of the iterative updating is derived from the fact that: a) as more matching entities are found, more relevant documents can be utilized for calculating or updating the IDF scores; b) as more correlative CText is put 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 of basic workflow, until the IDF scores become stable.

3.2 Cooccurrence-Based Sub-Topic Analytics Model

The baseline algorithm measures the similarity between two pieces of CText in one dimension only. However, as consolidated data, there is actually information of different sub-topics in each piece of CText. Different from those topics such as "sports", "music", "education" and so forth, the sub-topics can be taken as various aspects of the same topic. For instance, as for the house renting information there are several aspects of the information on "direction", "greening", "property", "traffic" and so forth for describing the situation of an apartment.

In this subsection, we introduce a novel algorithm that works on mining sub-topics from CText, and then calculate the similarity between CText on all sub-topic dimensions. Intuitively, if two phrases are always mentioned in same sentences, it is quite possible that there exists some association relationship between the two phrases. Based on this intuition, we will build up a phrase cooccurrence graph (PC-Graph), and then employ the so-called phrase association degree (PAD) to measure the similarity between two records on corresponding sub-topic dimensions.

1) *Constructing the PC-Graph.* Give a piece of CText ct, we divide it into a set of segments $t_1, t_2, ..., t_n$ according to the separators such as ",", ".", "?", stop-words and so on. We then employ the Longest-Cover method[10] to segment each segment for getting the longest terms in the given vocabulary after filtering the

stop-words. Next, we add edges with weights between every pair of phrases if the two phrases have co-occurred in the same segment, where weights of an edge between two phrases p_i and p_j can be calculated with the following formula:

$$freq(ct, p_i, p_j) = e^{-gap_{ct}(p_i,p_j)}\ bool(p_i, p_j),$$

where $gap_{ct}(p_i, p_j)$ presents the distance between p_i and p_j in the CText and $e^{-gap_{ct}(p_i,p_j)}$ is to penalize the long distance between two phrases, and $bool(p_i, p_j)$ is used to reduce the influence of similar phrases in the same CText which can be computed with the following formula:

$$bool(p_i, p_j) = \begin{cases} 1, & \text{if } sim(p_i, p_j) \leqslant \theta, \\ 0, & \text{otherwise,} \end{cases}$$

where the function $sim(\cdot,\cdot)$ computes the string similarity (e.g., edit similarity) between two phrases, and θ is the string similarity threshold.

Next, we count up the total frequencies of the cooccurence between the phrase pair (p_i, p_j) on all the CText in the training set denoted by T as follows:

$$Freq(p_i, p_j) = \sum_{ct \in T} freq(ct, p_i, p_j),$$

according to which we can calculate the PAD value of an edge linking p_i to p_j with the following formula:

$$PAD(p_i, p_j) = \frac{Freq(p_i, p_j)}{\sum\limits_{p \in \boldsymbol{P}_g} Freq(p_i, p)} \times \log \frac{|\boldsymbol{P}_g|}{|Adj(p_j)| - 1},$$

where $\frac{Freq(p_i,p_j)}{\sum_{p \in \boldsymbol{P}_g} Freq(p_i,p)}$ calculates the percentage of the degree between p_i and p_j for the total degree of p_i, $\log \frac{|\boldsymbol{P}_g|}{|Adj(p_j)|-1}$ is used to penalize a general phrase that always co-occurs with other phrases, $Adj(p_j)$ is a phrase set whose elements always occur with phrase p_j, and $|\cdot|$ gets the size of a set.

Example. Parts of the PC-Graph built on the house renting dataset are shown in Fig.3. As we can see, those phrase pairs that are always mentioned together will have a high PAD such as "convenience", "ease", and "southwest" with "traffic", while some phrase pairs that are only mentioned together once or twice will have a low PAD such as "good" with "traffic".

2) *Partitioning the PC-Graph.* As shown in Fig.3, there might be some weak association relationship (with low PAD scores) between phrase nodes, which prevent us from identifying topics from the graph. Thus, we now consider to divide the PC-Graph into graph partitions, expecting that each of the graph partitions will closely correspond to a topic. Inspired by the work/model in [11], our problem is translated into the following optimization problem: 1) maximizing the sum of PAD scores within each graph partition; 2) reducing the PAD scores across graph partitions. More formally, our problem is to maximize the following formula:

$$\text{maximize} \sum_{p_1 \in \boldsymbol{P}_g, p_2 \in \boldsymbol{P}_g, p_1 \neq p_2} \frac{PAD(p_1, p_2)}{dis(p_1) + dis(p_2) + \alpha}, \quad (2)$$

where

$$\begin{cases} dis(p_1) = \text{Max}_{p \in Adj(p_1)} PAD(p_1, p) - \\ \qquad\qquad \text{Min}_{p \in Adj(p_1)} PAD(p_1, p), \\ dis(p_2) = \text{Max}_{p \in Adj(p_2)} PAD(p_2, p) - \\ \qquad\qquad \text{Min}_{p \in Adj(p_2)} PAD(p_2, p), \end{cases}$$

where α is an equilibrium factor to prevent the denominator being 0.

Theorem 1. *Finding the optimal solution for* (2) *is an NP-hard problem.*

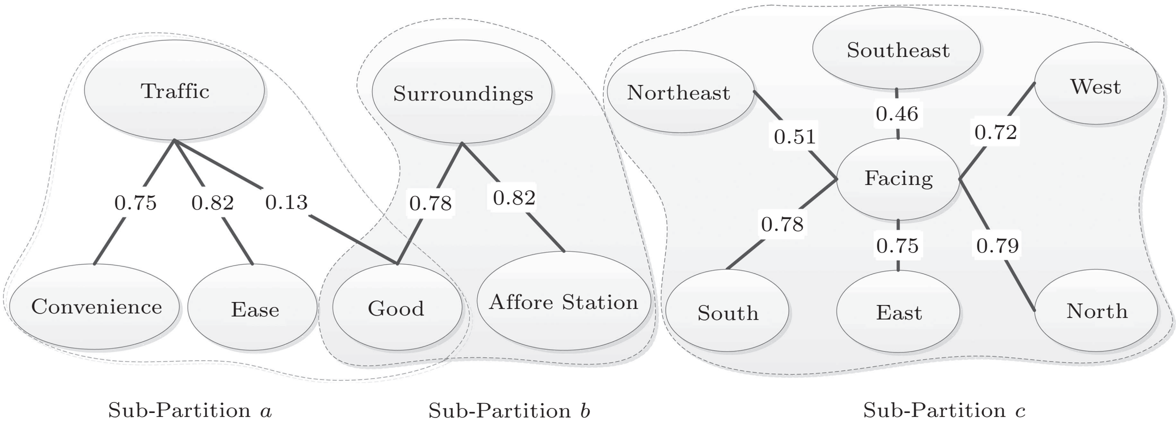


Fig.3. Example PC-Graph with expected three sub-partitions.

Proof. We prove that the optimal solution is NP-hard even if the number of micro-topics is given. We then prove it by the reduction from the balanced max-skip partitioning problem[12]. Given a set V of binary vectors, where $|V|$ is a multiple of p, we want to find a partitioning $\mathcal{P}$ over V such that the following total cost $\mathcal{C}(\mathcal{P})$ is maximized:

$$\mathcal{C}(\mathcal{P}) = \sum_{p_i \in \mathcal{P}} C(P_i),$$

where $C(P_i) = |P_i|$ is the cost of a graph partition P_i. In our case, we denote the cost of a graph partition P_i as

$$\begin{aligned} C(P_i) &= \sum_{p_1 \in \boldsymbol{P}_g, p_2 \in \boldsymbol{P}_g, p_1 \neq p_2} \frac{PAD(p_1, p_2)}{dis(p_1) + dis(p_2) + \alpha} \\ &= \sum_{p_1 \in \boldsymbol{P}_g, p_2 \in \boldsymbol{P}_g, p_1 \neq p_2} 1 - \Delta(P_i), \end{aligned}$$

where $\Delta(P_i)$ is similar to $\bar{v}(P_i)j$ in the balanced maxskip partitioning problem. Thus, (2) is equivalent to maximizing the total cost of $\mathcal{P}$, i.e., finding the optimal solution for (2) is NP-hard. □

As described in Theorem 1, it is hard to solve the non-linear optimization problem. In the following, we employ a greedy algorithm to solve the problem. Intuitively, we always greedily select the edge with the minimum PAD as the place to perform the partition.

We define a so-called cohesion score (CScore) of every graph partition G_p, which can be calculated with the following equation:

$$\begin{aligned} & CScore(G_p) \\ = & \sum_{(p_1,p_2) \in \boldsymbol{P}_{G_p}} PAD(p_1, p_2) / (\max_{(p_1,p_2) \in \boldsymbol{P}_{G_p}} PAD(p_1, p_2) - \\ & \min_{(p_1,p_2) \in \boldsymbol{P}_{G_p}} PAD(p_1, p_2) + \alpha), \end{aligned}$$

where α is an equilibrium factor to prevent the denominator being zero, and $\boldsymbol{P}_{G_p}$ denotes the set of phrases in the partition G_p. Assume that the graph partition G_p will be divided into two sub-graph partitions G_{p1} and G_{p2} at the edge with the minimum PAD. If this partition operation satisfies the following conditions (3), we will carry out the partition operation.

$$\begin{cases} CScore(G_p) \leqslant CScore(G_{p1}) + CScore(G_{p2}), \\ |CScore(G_{p1}) - CScore(G_{p2})| \\ \leqslant \min\limits_{(p_1,p_2) \in \boldsymbol{P}_{G_p}} PAD(p_1, p_2), \\ |G_{par1}| > 1, |G_{par2}| > 1. \end{cases} \quad (3)$$

For each graph partition, we iteratively select an edge with the minimum PAD to divide the partition until no more edges satisfy the conditions listed in (3).

3) *Acquiring Sub-Topics and Weights.* We now acquire sub-topics from the graph partitions. For every graph partition, we calculate an average PAD score for every node in the partition, and then select the one with the highest average PAD score as the sub-topic phrase. Then we take all the other phrases that co-occur with the sub-topic phrase as the sub-topic values.

Assume we get K sub-topics denoted in a vector $(subT_1, subT_2, ..., subT_K)$ from the PC-Graph, where each $subT_i$ $(1 \leqslant i \leqslant K)$ denotes a sub-topic. For every dimension, we employ domain knowledge to set weights of different sub-topics for matching, whose identification degree is in the form of a weight vector $(w_1, w_2, ..., w_K)$. Initially, we set $w_k = 1$ $(1 \leqslant k \leqslant K)$, but the weights will be updated iteratively with the entity matching results changing. According to the entity matching result after an iteration, we update the weight w_i as described in (4). We iteratively update the weight vector until it becomes stable.

$$w_k = \frac{Pos_{subT}(k)}{Pos_{subT}(k) + Neg_{subT}(k)}, \quad (4)$$

where $Pos_{subT}(k)$ is the number of all entity pairs (r_i, r_j) satisfying: if $r_i[k] = r_j[k]$, the entity pair (r_i, r_j) is linked in the current iteration, while $Neg_{subT}(k)$ is the number of all entity pairs (r_i, r_j) satisfying: if $r_i[k] = r_j[k]$, the entity pair (r_i, r_j) is not linked in the current iteration.

4) *Matching Entities on Sub-Topics.* Initially, we identify the sub-topic for every CText segment of every record. We then calculate the similarity between every record pair in one block r_i and r_j with the adjusted cosine similarity function[13] as follows:

$$Sim(r_i, r_j) = \frac{\sum_{k=1}^{K} w_k^2 \times sim(r_i[k], r_j[k])}{\sum_{k=1}^{K} (w_k \times sim(r_i[k], r_j[k]))^2}.$$

However, it may happen that there is no obvious sub-topic phrase in a CText segment, which leads to fail to directly identify the sub-topic of the segment. In this case, we employ a probabilistic model to deduce the probability of which topic it belongs to. Let $\boldsymbol{P}(t)$ denote the set of phrases identified in the segment t, and we use the following (5) to calculate the probability that t belongs to a sub-topic $subT$ according to the law of total probability.

$$Pr(subT|\boldsymbol{P}(t))$$

$$= \sum_{p \in \boldsymbol{P}(t)} \frac{Pr(p|subT) \times Pr(subT)}{\sum_{subT} Pr(p|subT) \times Pr(subT)}, \qquad (5)$$

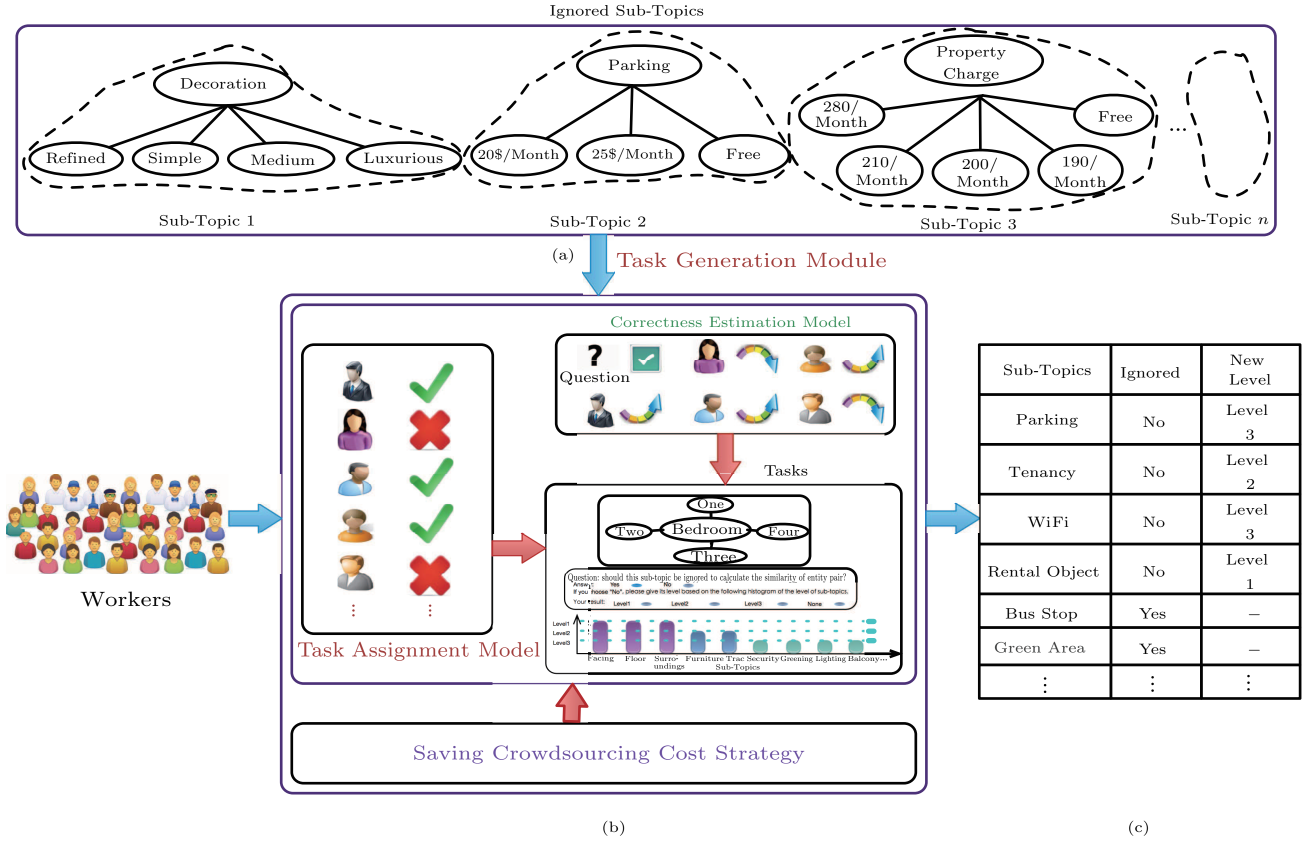
where $subT$ is a sub-topic, and $Pr(p|subT)$ is the probability that phrase p occurs on the condition of topic $subT$, which can be calculated by our prior knowledge.

After calculating the probability that t belongs to different sub-topics, we select the maximum probability one as the sub-topic of the segment, and then still use (5) to calculate the similarity between two records.

4 Crowd-Guided CTextEM with CText

One weakness of the IDF-based approach lies on its ability in estimating the importance of different sub-topics in EM. In other words, it may underestimate some important hidden sub-topics and thus decrease the recall of EM. For example, weights of sub-topic “tenant type” might be very low for the “House Renting Information” dataset, but it is actually an important factor in doing EM on this dataset. Given this, we propose to use the crowd as our guidance to help adjust weights of different sub-topics in EM. For this purpose, we work on estimating the accuracy of each worker and selecting the most suitable worker to fulfill generated tasks.

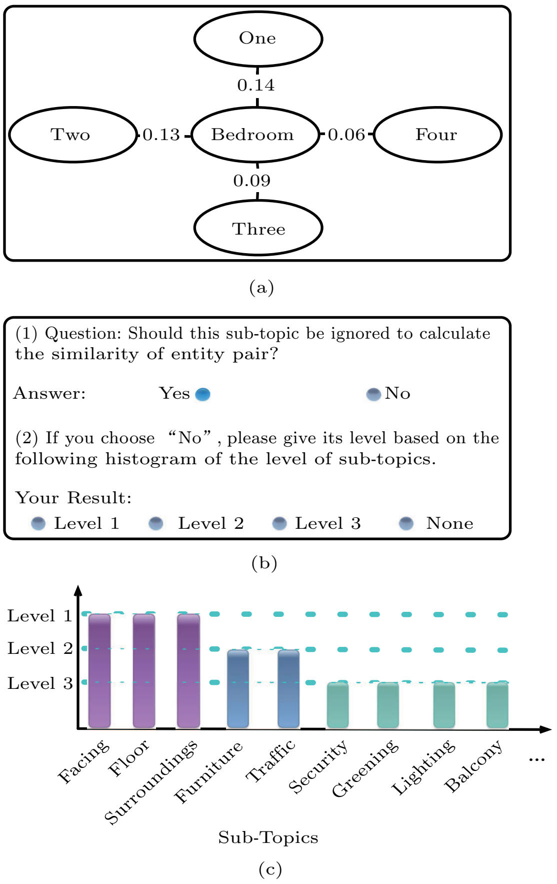
We describe the workflow of the crowd guidance model in Fig.4, which mainly consists of four modules. 1) The task generation module is responsible for generating tasks based on our input sub-topics which are presented in the form of PC-Graphs as shown in Fig.4(a) (see details in Fig.5). Each task has two questions as depicted in Fig.5. The first question is a binary one, which must be answered. If the answer to the first question is “Yes”, then the second question will be skipped; otherwise, the second question also needs to be answered. The second question has four options, “level 1” (if the workers think the weight of the sub-topic w is larger than 0.76 in our experiments, i.e., $w > 0.76$), “level 2” (if $0.76 \leqslant w < 0.51$), “level 3” (if $0.51 \leqslant w < 0.24$), and “none” which means the worker has no idea about the weight of the sub-topic. The weight bounds we set here are acquired by emulating the idea of cluster algorithm and based on the distribution of weights of the sub-topics. We refer to



Sub-Topics	Ignored	New Level
Parking	No	Level 3
Tenancy	No	Level 2
WiFi	No	Level 3
Rental Object	No	Level 1
Bus Stop	Yes	–
Green Area	Yes	–
⋮	⋮	⋮

Fig.4. Workflow of the crowd guidance model in CTextEM. (a) Some ignored sub-topics. (b) Crowd guidance model. (c) Final results of workers. The detail of the example task for crowdsourcing is shown in Fig.5.

the weights of those unignored sub-topics to acquire the corresponding levels to infer ranges for these ignored sub-topics due to the uniform distribution. 2) The correctness estimation model not only calculates the probability that an option is the right answer to its corresponding question, but also estimates the accuracy of each worker. 3) Task assignment model helps the correctness estimation model select the most suitable workers to fulfill tasks. 4) Saving crowdsourcing cost model aims at saving the cost of crowd intervention in the condition of having the similar crowdsourcing accuracy.



(a)

(1) Question: Should this sub-topic be ignored to calculate the similarity of entity pair?

Answer: Yes ● ● No

(2) If you choose "No", please give its level based on the following histogram of the level of sub-topics.

Your Result:
● Level 1 ● Level 2 ● Level 3 ● None

(b)

(c)

Fig.5. Example task for crowdsourcing. (a) Ignored sub-topic. (b) Task: two step questions. (c) Candidate weights for ignored sub-topics.

In the rest of this section, we first introduce the question-worker estimation model in Subsection 4.1, and then present the worker selection model in Subsection 4.2. We finally give a way to save the crowdsourcing cost in Subsection 4.3.

4.1 Correctness Estimation Model

The correctness estimation model calculates the correctness of answers and the confidence of workers simultaneously since the two depend on each other. Given that each of our generated tasks has two questions, we ask that each worker must answer the first question and can skip the second. Based on this assumption, we can build our question-worker estimation model with the first binary question only. In the rest of this subsection, we use "question" to denote our binary question only.

1) *Worker Accuracy Estimation.* The correct probability of an answer to a question is mainly decided by the percentage of workers supporting this answer, and the reliability (i.e., confidence) of every worker. Some previous methods[14] simply estimate the reliability of workers according to the average accuracy of a worker's answers to questions in history, which is not accurate enough. In this paper, inspired by Feng *et al.*[14], we introduce a 0-1 confusion matrix model to estimate the reliability of a worker $u_k \in U$ (U denotes the worker set), denoted by $\boldsymbol{W}_k$, as follows:

$$\boldsymbol{W}_k = \begin{pmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{pmatrix},$$

where each c_{xy} ($x, y \in \{0, 1\}$) denotes a different status about the worker's answer and the consensus answer to a question. For instance, for a question $q_i \in Q$ (Q is the question set), if the consensus answer, say a_i^*=1, and u_k's answer, say a_i^k=1 (where "1" stands for that the worker thinks the answer is right while "0" represents the worker considers the answer wrong), then $c_{00_i} = 0$, $c_{01_i} = 0$, $c_{10_i} = 0$, and $c_{11_i} = 1$. That is, we evaluate the answers of workers based on the voted consensus answers. For the i-th question, if the workers' answer is the same with the consensus answer, it generates a positive contribution; otherwise, a negative contribution. We show the complete relationship with the following equation:

$$\begin{cases} c_{00_i} = 1, c_{01_i} = c_{10_i} = c_{11_i} = 0, \text{if } a_i^k = a_i^* = 0, \\ c_{11_i} = 1, c_{00_i} = c_{10_i} = c_{01_i} = 0, \text{if } a_i^k = a_i^* = 1, \\ c_{10_i} = 1, c_{00_i} = c_{01_i} = c_{11_i} = 0, \text{if } a_i^k = 1, a_i^* = 0, \\ c_{01_i} = 1, c_{00_i} = c_{10_i} = c_{11_i} = 0, \text{if } a_i^k = 0, a_i^* = 1. \end{cases}$$

Straightforwardly, the average accuracy of the answer given by the worker u_k can be estimated by:

$$\gamma_k = \frac{\sum_{i \in [1, N_k]} c_{00_i} + \sum_{i \in [1, N_k]} c_{11_i}}{\sum_{x=\{0,1\}, y=\{0,1\}} \sum_{i \in [1, N_k]} c_{xy_i}},$$

where N_k is the number of questions answered so far.

However, we cannot just use γ_k to denote the reliability of a worker since it neglects the number of questions a worker has answered and as a result, some "lucky" fresh workers who always give a random but right answer to questions may be assigned more tasks. Therefore, by taking the number of questions a worker has answered denoted by n_k, we have a more reliable estimation method and the confidence of worker u_k is defined as follows:

$$c(u_k) = e^{-\frac{1}{1+n_k}} \times \gamma_k. \tag{6}$$

The curve of (6) is also given in Fig.6. As can be observed in the figure, when a fresh worker randomly gives right answers a few times, although the average answer accuracy γ_k becomes high, the worker's reliability $c(u_k)$ increases slowly. Fig.6 demonstrates that (6) properly solves the problem that the "lucky" fresh workers randomly give right answers. With these workers answering more questions, the probability of "luck" is greatly weakened and the confidence of workers providing right answers trends to be their true capability. This curve demonstrates this fact well.

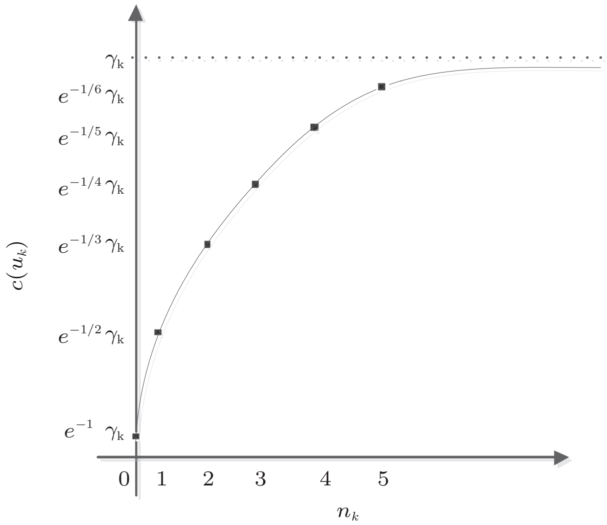


Fig.6. Curve of (6).

2) *Answer Correctness Estimation.* Initially, we set a unified correctness probability, say 0.5, for each answer to its corresponding question, such that we can estimate the worker accuracy accordingly. After estimating the accuracy of workers with a number of tasks, we can then update the correctness of each answer a to a question in task t as follows:

$$c(t, a) = \frac{\sum_{u_k \in U(t,a)} c(u_k)}{\sum_{u_k \in U(t)} c(u_k)},$$

where $U(t)$ denotes the set of workers that have been assigned task t and $U(t, a)$ denotes workers that have given answer a to task t.

Each time after we update the correctness of answers, we will also update the confidence of workers accordingly by updating (6) with the below equation:

$$\gamma_k = \frac{\sum\limits_{i \in [1,n_k]} (c_{00_i} \times c(t_i, 0)) + \sum\limits_{i \in [1,n_k]} (c_{11_i} \times c(t_i, 1))}{\sum\limits_{x \in \{0,1\}, y \in \{0,1\}} \sum\limits_{i \in [1,n_k]} (c_{xy_i} \times c(t_i, y))}.$$

4.2 Task Assignment Model

The task assignment model needs to reduce the cost of crowdsourcing for accomplishing all tasks while ensuring the high quality of the answers. In order to reach this goal, we mainly need to deal with two issues. 1) What kind of workers should we assign new tasks to? 2) When should we stop assigning more workers to a particular task?

For the first issue, we tend to keep a free worker pool which contains all workers that are free at this moment, and will be updated in time according to the state of workers. Each time when we need a new worker for a specific task, we greedily choose a worker with the highest confidence (i.e., the $c(u_k)$ calculated by (6)) from the free worker pool, until a stopping condition is met.

The second issue is more challenging for it depends on not only the answer given by every worker assigned task t, but also the confidence of the worker. Here we continue to let $U(t)$ denote the set of workers that have been assigned task t and $U(t, a)$ denote the workers that have given answer a to task t. We can simply estimate the correctness of each answer a to task t according to the noisy-all model as follows:

$$c(t, a) = 1 - \prod_{u \in U(t,a)} (1 - c(u)).$$

We will stop assigning more workers to task t if one of the following two stopping conditions is met: 1) there is an answer a which satisfies that $c(t, a)$ is larger than a predefined threshold θ_c; or 2) the number of workers participating in the task t reaches the maximum number of workers we set for a task (e.g., we set 20 as the maximum number of crowd-workers for one task based on our experimental results).

Example. Given task t, assume that θ_c is 0.95. Table 1 shows the workers in the working pool and their corresponding confidences and answers. Based on the task assignment model and Table 1, we first assign u_4 to task t. Thus we have $c(t, a) = 1-(1-0.8) = 0.8 < \theta_c = 0.95$, and then continue to assign u_3 to the task. Finally, we have $c(t, b) = 1-(1-0.75) = 0.75 < \theta_c = 0.95$.

We continue to assign u_1 to the task and have $c(t, a) = 1 - (1 - 0.8)(1 - 0.7) = 0.94 < \theta_c = 0.95$. Next, we get u_2 to the task and we have $c(t, c) = 1 - (1 - 0.65) = 0.65 < \theta_c = 0.95$. And then we select u_5 to the task and then have $c(t, a) = 1-(1-0.8)(1-0.7)(1-0.63) = 0.978\,4 > \theta_c = 0.95$. Until now, we can stop assigning more workers to the task, and the answer to the task is a whose correctness is 0.978 4.

Table 1. Workers in the Working Pool

Worker	Confidence	Answer
u_1	0.70	a
u_2	0.65	c
u_3	0.75	b
u_4	0.80	a
u_5	0.63	a
u_6	0.58	b

4.3 Saving Crowdsourcing Cost Model

Since there can be a large number of sub-topics generated from a dataset, if we let the crowd decide whether each of these sub-topics should be ignored or not, the cost of crowd intervention could be very high. To save the crowdsourcing cost, we find an approximate way to reduce the number of crowd intervention.

Recall that each generated sub-topic has a weight to denote its identification degree in doing CTextEM as mentioned in Subsection 3.2, and this weight can reflect the importance of every sub-topic to some extent. Thus, if a number of, say, K_b adjacent sub-topics with similar weights are decided to be ignored by crowdsourcing, other sub-topics with lower weights should also be ignored. Similarly, if K_b adjacent sub-topics are treated as accepted ones by crowdsourcing, other sub-topics with higher weights should also be regarded as accepted ones. Based on this intuition, we hope to find the lower-bound and upper-bound for the weights of sub-topics that need to be checked by crowdsourcing. Here we usually set $K_b = 3$ which has been observed in our experiments that can ensure the statement above is safe.

Inspired by the Binary Search algorithm, we perform lower-bound and upper-bound searching algorithm as follows. Initially, we sort all sub-topics based on their weights in descending order. We then set the minimum weight *min* as the start position weight *start*, the maximum weight *max* as the end position weight *end*, and the medium weight *med* as the middle position *mid* such that all the sub-topics and their weights can be divided into three parts. Next, we generate tasks for sub-topics at positions *start*, *mid* and *end*.

- If K_b adjacent sub-topics at position *start* are ignored, the sub-topics whose weights are lower than it should also be ignored.
- If K_b adjacent sub-topics at position *end* are accepted, the sub-topics whose weights are larger than them should also be accepted.
- If K_b adjacent sub-topics at position *mid* are ignored, the sub-topics whose weights are between (including) *min* and *mid* should be ignored. Or, if K_b adjacent sub-topics at position *mid* are accepted, the sub-topics whose weights are between (including) *mid* and *max* should be accepted.

For the remaining sub-topics, we further update the medium weight *med* as the new middle position *mid*, the minimum weight *min* as the start position weight $mid + 1$ (or, $min + 1$) and the maximum weight as the end position weight $end - 1$ (or, $mid - 1$), and iteratively go on the above steps until $min \leqslant max$ or no more remaining sub-topics.

As we describe above, we first employ the Quick Sort algorithm to sort these sub-topics based on their weights and then use the idea of the Binary Search algorithm to select which sub-topics should be intervened by workers. Therefore, the time complexity of our proposed algorithm is $O(|Q| \times \log |Q| + \log |Q|)$, where Q is the set of questions.

5 Experiments

We implement all the methods with Java and our experiments are run on a PC with Intel core i5 duo 2.6 GHz CPU and 8 GB RAM. For the crowdsourcing tasks, we perform our experiments on the Amazon Mechanical Turk platform.

5.1 Datasets and Metrics

Our experiments are performed on two real-world datasets collected by ourselves from the Web.

- *House*. This database contains the house renting information collected from three house renting information websites, Ganji[1], Anjuke[2], 58tongcheng[3] of five

[1] http://ganji.com, Aug. 2017.

[2] http://www.anjuke.com, Aug. 2017.

[3] http://www.58.com, Aug. 2017.

large-medium cities of China: Beijing, Shenzhen, Tianjin, Chengdu, Suzhou. The property of the database is given in Table 2.

Table 2. Information of the Two Datasets: House and Car

Dataset	Source	Number of Attributes	Number of Records ($\times 10^3$)
House	Beijing	22	5.6
	Chengdu	22	8.6
	Suzhou	22	10.8
	Shenzhen	22	17.1
	Tianjin	22	13.5
Car	Toyota	12	5.6
	Audi	12	5.2
	BMW	12	6.0
	Honda	12	5.5
	Buick	12	5.8

- *Car*. This database contains second-hand cars for selling crawled from Ganji website and "che168" website, which contains the information of second-hand cars of several brands including Toyota, Audi, BMW, Honda and Buick. The property of this dataset is also given in Table 2.

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 $F1 = \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$. We use the time cost of an algorithm for evaluating the efficiency of a method. Besides, we also evaluate the effect of the accuracy threshold θ_c and the effect of our proposed crowd-guided CTextEM algorithm on the $F1$ score and the needed number of workers.

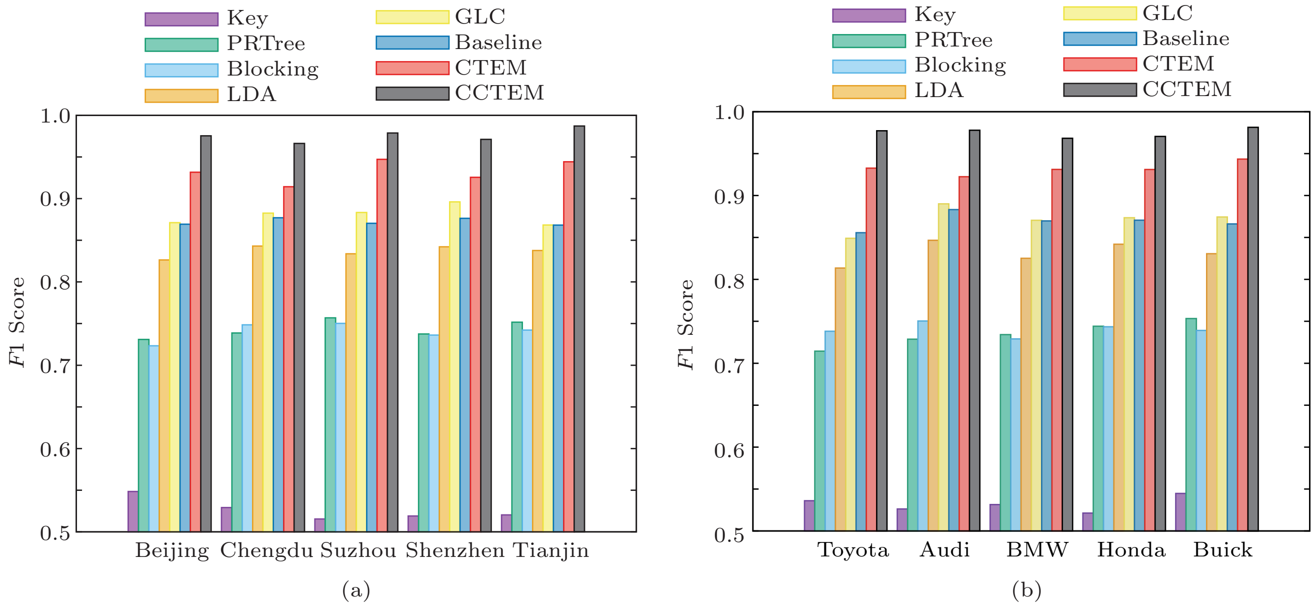
5.2 Comparison with Previous Methods

In this subsection, we compare the effectiveness of our three CTextEM algorithms, i.e., Baseline (IDF-based EM), sub-topic based method (CTEM) and CrowdGuided CTEM (CCTEM), with several state-of-the-art EM methods and also CText-based EM methods by using other classical topic-models.

- The key-based EM method integrates many state-of-the-art techniques based on key values for reducing the comparison cost, such as Q-gram[15] and inverted indices[16].
- The blocking-based EM method[17] selects some attributes with high identification to create hash buckets for matching entities. The entities in the same buckets are likely to be the same, while the entities with different hash codes cannot be the same.
- The PRTree-based EM method[18] builds up a probabilistic rule-based decision tree based on all attributes such that they can perform efficient and effective EM with both key and non-key attributes.
- The LDA-based EM method relies on the LDA topic model[7] to mine the hidden variables named topics from CText to build up topic vectors for calculating the similarities.
- The GLC-based EM method relies on the GLS topic model[19] to understand the information in CText and then builds up topic vectors for calculating the similarities.

As shown in Fig.7(a), relying on key attributes only, the key-based EM has the lowest $F1$ scores. The effect of blocking-based EM is discounted greatly due to the missing values of the structured data, which leads to the occurrences of false-positive. PRTree EM works better than the key-based EM method but worse than the Baseline, since PRTree uses non-key structured attributes but does not use CText. Our baseline algorithm extracts information from CText combining with structured data to do EM, and thus reaches a higher $F1$ score. The accuracy of LDA-based EM is lower than that of Baseline, because it is not good at learning sub-topics from CText. The Baseline EM and GLC-based EM are very close, but they are both worse than our sub-topic method since our sub-topic method uses the CText information in an advanced way. After all, the CCTEM method gets the best $F1$ score among all methods. It uses workers to reselect the ignored sub-topics and then utilizes the union of sub-topics with high weights and accepted sub-topics to do EM based on the sub-topic method. Therefore, it can capture more important information from CText data to help us find more matched entity pairs.

For more comprehensive comparison, we compare the precision and recall of these methods on the House dataset. As listed in Table 3, CCTEM gets the highest precision and recall among all methods (the last row in bold). Because of the usage of crowdsourcing, the ignored sub-topics with low weights but playing an important role in EM are found and more entity pairs which were not be matched before are matched together with our CCTEM algorithm. Thus CCTEM gets the best effectiveness. And the sub-topic EM (CTEM) reaches the second highest precision and recall compared with others (the last line but one in bold), while GLC-based EM is worse than CTEM only at the third



(a)

(b)

Fig.7. Compared with previous methods on $F1$ score. (a) The House dataset of five cities. (b) The Car dataset of five brands.

Table 3. Comparison with Previous Methods on Five Cities of the House Dataset

Method	Beijing		Chengdu		Suzhou		Shenzhen		Tianjin	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Key	0.699 4	0.451 2	0.711 6	0.421 2	0.725 4	0.399 8	0.705 9	0.410 5	0.714 2	0.409 3
PRTree	0.750 4	0.712 5	0.754 2	0.723 9	0.755 6	0.758 2	0.769 4	0.708 1	0.756 2	0.747 0
Blocking	0.745 2	0.702 8	0.764 5	0.733 2	0.758 3	0.742 5	0.746 7	0.725 9	0.755 6	0.729 3
LDA	0.847 2	0.806 6	0.861 6	0.825 3	0.843 8	0.824 1	0.852 7	0.832 0	0.845 5	0.830 2
GLC	0.880 1	0.862 5	0.896 4	0.869 3	0.904 5	0.863 2	0.936 6	0.859 0	0.884 7	0.852 6
Baseline	0.896 6	0.843 7	0.905 9	0.849 8	0.889 1	0.852 4	0.910 5	0.844 7	0.872 5	0.863 9
CTEM	**0.968 8**	**0.897 4**	**0.947 2**	**0.883 6**	**0.980 2**	**0.916 3**	**0.965 0**	**0.889 2**	**0.982 3**	**0.908 9**
CCTEM	**0.984 7**	**0.936 2**	**0.963 4**	**0.949 1**	**0.987 4**	**0.970 3**	**0.985 2**	**0.957 4**	**0.990 6**	**0.953 7**

highest precision and recall. The effect of the Baseline method is similar to the GLC-based EM, but the LDA-based EM is the worst of the four methods using CText.

5.3 Evaluating the Extracted Results from CText

We compare the key information extracted from CText with different topic models and our methods. As shown in Table 4, our cooccurrence-based sub-topic analytics model can acquire more accurate information than others with the aid of the sub-topic vectors we generated. However, the LDA model only gets some information roughly as shown in the table which is not accurate enough for EM. As can be observed in the table, some important phrases such as "Community Planning" are divided into two phrases. The GLC model cannot get sub-topics or sub-topic phrases well. For example, the phrases "floor" and "twenty" are mixed together. The results of Baseline EM are similar to those of GLC-based EM. Both of them get good performance. However, we can see that other methods except the CCTEM method do not find the ignored sub-topics, such as "tenant type". All in all, the cooccurrence-based sub-topic analytics model is more suitable than other models for understanding the information of CText without crowdsourcing. By employing crowdsourcing, the CCTEM algorithm gets better results than the cooccurrence-based sub-topic analytics model.

We also list the weights of different sub-topics on the House dataset in Table 5. As can be observed, the sub-topic "floor" has a higher weight than the others since it can better decide the matching results on the dataset. It is consistent with our exception that the

Table 4. Comparison of Extracted Information of Different Models

Method	An Example of CText	Another Example of CText
	Community planning well, unique warmth, flowers and trees patchwork, like a garden, furniture and appliances equipped well, refined decoration, facing south right, twenty floor, tenant types limit for family, ...	South facing, good lighting, two air conditioning, water heaters and washing machines equipped, free of property charges, ...
LDA	Community, planning, warmth, flowers, trees, garden, furniture, appliances, decoration, south, floor, tenant types, family, ...	South, facing, lighting, air, conditioning, water, heaters, washing, machines, property, charges, ...
GLC	Community planning, warmth, flowers and trees, garden, furniture and appliances, refined, decoration, south, twenty, floor, tenant types, family, ...	South, lighting, two, air conditioning, water heaters, washing machines, free, property charges, ...
Baseline	Community planning, well, warmth, flowers and trees, garden, furniture and appliances, refined, decoration, south, facing, floor, tenant types, family, ...	Facing, south, lighting, air conditioning, water heaters and washing machines, property charges, ...
CTEM	Community planning, warmth, flowers, trees, furniture, appliances and decoration, well-groomed, facing, floor, tenant types, family, ...	Facing, lighting, air conditioning, water heaters, washing machines, property charges, ...
CCTEM	Community planning, warmth, flowers, trees, furniture and appliances, decoration, well-groomed, facing, floor, tenant types, family, ...	Facing, lighting, air conditioning, water heaters, washing machines, property charges, ...

micro-topic with a higher identification degree owns a larger weight than the others. We also list some new sub-topics that are founded by the CCTEM method in Table 6. We can see that some important sub-topics are found which are ignored by the cooccurrence-based sub-topic analytics model.

Table 5. Example Sub-Topics Found by CTEM for the House Dataset

Phrase	Weight
Furniture and appliances	0.75
Decoration	0.69
Color	0.44
Facing	0.85
Floor	0.89
⋮	⋮

Table 6. Example Sub-Topics Identified by Crowdsourcing for the House Dataset

Phrase	Weight
Rental object	0.76
Parking	0.24
Tenancy	0.51
Owner	0.76
WiFi	0.24
⋮	⋮

5.4 Scalability Evaluation

We compare the $F1$ score and the time cost of the Baseline, CTEM and CCTEM methods with those of previous topic models like LDA and GLC. As illustrated in Fig.8(a), as the number of records increases from 100 to 10 000, the $F1$ scores of CTEM and CCTEM are very stable and always higher than those of the other compared methods. Besides, with the help of crowdsourcing, the $F1$ score is improved further with CCTEM on the foundation of the CTEM method given that the ignored sub-topics are found and used to calculate the similarity. In Fig.8(b), we can see that the time cost of sub-topic EM is also always less than that of the Baseline and the other topic models. We do not compare the time cost of CCTEM with the other methods since it needs the participation of workers whose time cost is with uncertainty.

5.5 Accuracy Threshold Evaluation

We evaluate the influence of accuracy threshold θ_c on $F1$ score of CCTEM and the number of required workers. As we can see in Fig.9, when the accuracy threshold θ_c increases, the $F1$ score of CCTEM also goes up with it on the two datasets. We find that the higher the accuracy threshold θ_c is, the more accurate the sub-topics are. Besides, we find that without the Worker Selection Model, the $F1$ scores just reach about 0.91 for the House dataset and 0.92 for the Car dataset, since poor-quality workers contribute to erro-

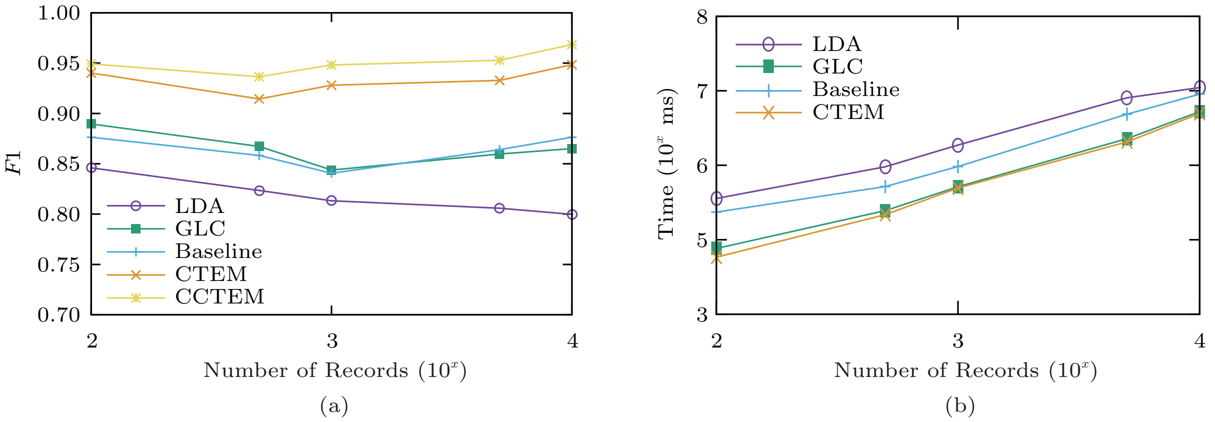


Fig.8. Comparison with previous topic model EM methods on (a) $F1$ score and (b) time cost. x is abscissa value.

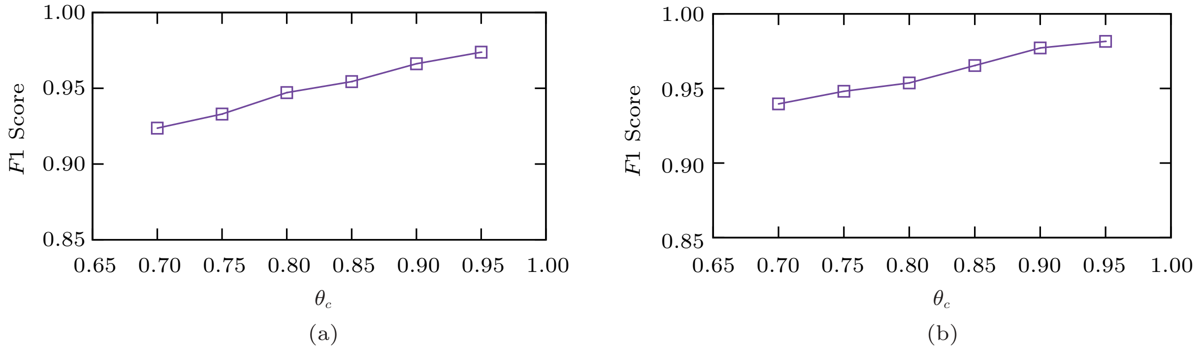


Fig.9. Relationship between accuracy threshold θ_c and $F1$ score of the CCTEM algorithm on the two datasets. (a) House dataset. (b) Car dataset.

neous answers. As illustrated in Fig.10, when the accuracy threshold θ_c varies, the number of required workers also changes. But when θ_c reaches 0.96 in Fig.10(a) and 0.9 in Fig.10(b), the number of workers keeps stable. In the first stage, because of the increase of accuracy, it needs more workers to provide answers. And in the second stage, the number of workers remains unchanged since our Worker Selection Model does not select workers any more. We find if workers with low accuracy fulfill tasks in the second stage, the accuracy will be harmed instead.

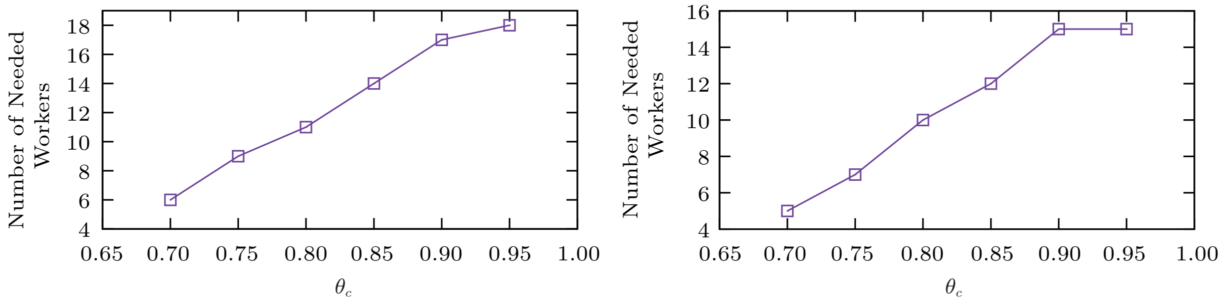
5.6 Crowd Cost Saving Evaluation

In this subsection, we evaluate the effect of our crowd cost saving algorithm on two aspects: $F1$ score and the number of crowdsourcing interventions. As we can see in Fig.11, the CCTEM algorithm with the cost saving algorithm (denoted by CCTEMSaving) has a similar $F1$ score with the CCTEM algorithm without cost saving (denoted by CCTEMNoSaving). The reason is that although some sub-topics have low weights, they are important in fact. This results in a small loss of $F1$ score (no more than 0.01), but it greatly decreases the number of crowdsourcing intervention by about 50% shown in Fig.12.

6 Related Work

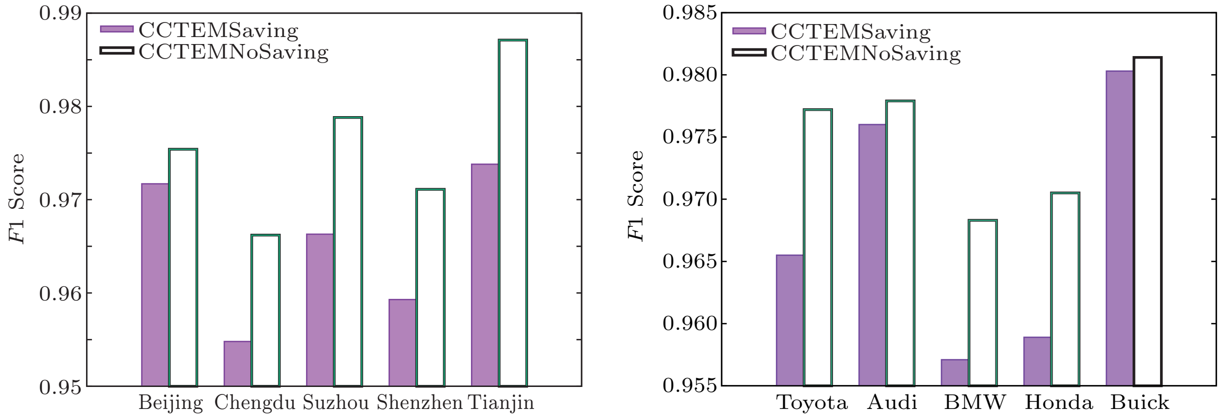
So far, plenty of work has been done on EM based on the string similarities[1], correlations[2], or semantic similarity[20] between various kinds of structured attribute values of the records such as digital values, date values or short string values in EM (see [3] for a survey). However, EM based on structured information only may easily fail when the structured information is not enough to identify the matching relationships between records.

As a complement 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 piece of CText, the conventional string similarity metrics cannot be applied directly. To utilize the information in CText for EM, the key is to identify



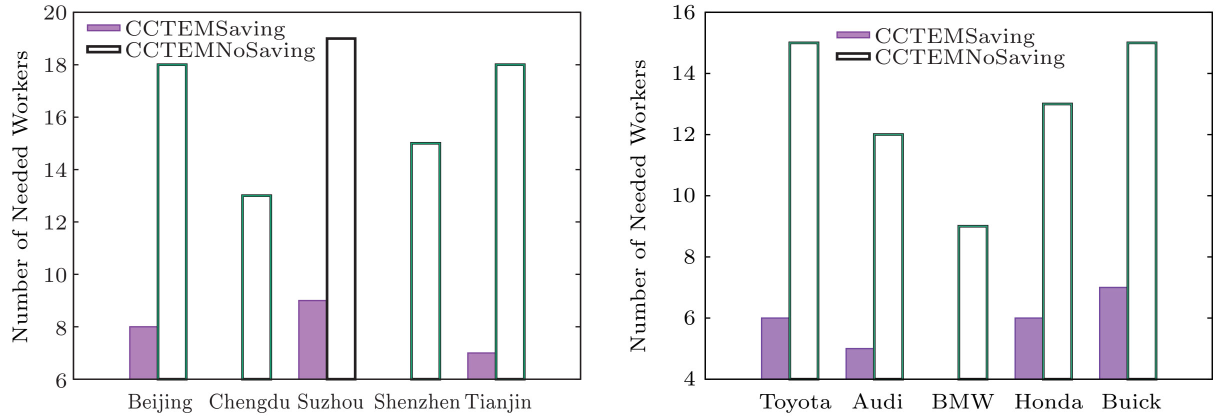
(a) (b)

Fig.10. Relationship between accuracy threshold θ_c and the number of needed workers of the CCTEM algorithm on the two datasets. (a) House dataset. (b) Car dataset.



(a) (b)

Fig.11. Comparison between CCTEMSaving and CCTEMNoSaving on $F1$ score on the two datasets. (a) House dataset. (b) Car dataset.



(a) (b)

Fig.12. Comparison between CCTEMSaving and CCTEMNoSaving on the number of needed workers on the two datasets. (a) House dataset. (b) Car dataset.

useful information from noises, and a big challenge is how to identify the key information[21]. Recently, some work has been done for unstructured information. A model based on unstructured text was present in [22], which arrives at a good precision and recall demonstrated with DBWorld posts. However, it needs the support of a special ontology largely. Besides, Ektefa *et al.*[4] considered a combination of string similarity and semantic similarity between two records, but the measure is not robust since 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 datasets.

There are also some researches on text understanding. Zhang and LeCun[23] applied deep learning to text understanding from character level inputs to abstract text concepts, using temporal convolutional networks. They devoted to learning about the main idea of CText rather than considering the relationship among phrases from CText. Besides, there are some topic models algorithms to discover the main themes for text information in the field of NLP (natural language processing), such as LDA[7], LSA[8] and PLSA[9]. They can get the hidden variables named topic words from texts. However, these methods will fail without the obvious topic of texts to get the useful information from CText. And some literatures about sub-topic mining have been proposed. Kim and Lee[24] proposed a method using the co-occurrence of words based on the dependency structure, and anchor texts from web documents to mine sub-topics. But the result of this method is limited by the quality of query and must be supported by external resources. Wu *et al.*[25] combined LDA and co-occurrence theory to determine text topics. However, it needs to be interpreted by experts to learn about the topics distribution of texts.

Recently, crowdsourcing has attracted significant attention in both industrial and academic communities[26-28]. Li *et al.*[26] reviewed extensive studies on crowdsourced data management. They reviewed existing methods on balancing quality, cost and latency and gave corresponding techniques for above problems. Besides, they showed some existing crowdsourced data management systems and optimization techniques. Li *et al.*[29] developed a crowd-powered database system CDB which adopts a graph-based query model to perform the multi-goal optimization. They focused on how to formulate the task selection problem, how to reduce latency, and how to optimize the quality for optimizing queries.

There are already researches embedding Crowdsourcing in EM[30-33]. For example, Wang *et al.*[31] proposed a hybrid human-machine approach for entity resolution in which machines are used to do an initial coarse pass over all the data, and workers are used to verify only the most likely matching pairs. Gokhale *et al.*[34] proposed a hands-off crowdsourcing for entity matching, which crowdsources the entire workflow of a task without developers. Demartini *et al.*[33] proposed a new approach to combining named entity disambiguation, coreference resolution and alias detection with crowdsourcing-based CR. They first built semantic markup for entities from the web of unstructured contents and then used annotated contents to improve above automated methods with crowdsourcing. Our work is different from the existing ones in that we just employ crowdsourcing to fulfill a part of TextEM, and then improve the effectiveness of EM with the most suitable workers.

7 Conclusions

We worked on employing CText in EM, i.e., the CTextEM problem, in this paper. To solve the problem, we proposed a novel cooccurrence-based topic model to identify various sub-topics from each piece of CText, and then measured the similarity between CText on the multiple sub-topic dimensions. We also let the crowd to help improve the sub-topic identification model. Extensive experimental study based on several data collections demonstrated that our proposed Cooccurrence-based Sub-Topic Analytics model can effectively identify sub-topics from CText and thus help improve the accuracy of EM in average 10% of the Iterative IDF-Based CTextEM algorithm. In addition, crowdsourcing can further improve the accuracy of EM about 5% on average.

As future work, it would be interesting to use knowledge graph to help us do EM that can improve the accuracy further, like domain-based knowledge graph. It would also be interesting to investigate whether employing web to mine useful information can improve the matching results of our proposed methods.

References

[1] Koudas N, Sarawagi S, Srivastava D. Record linkage: Similarity measures and algorithms. In *Proc. the ACM SIGMOD Int. Conf. Management of Data*, June 2006, pp.802-803.

[2] Parkhomenko E, Tritchler D, Beyene J. Sparse canonical correlation analysis with application to genomic data integration. *Statistical Applications in Genetics and Molecular Biology*, 2009, 8(1): Article No. 1.

[3] Elmagarmid A K, Ipeirotis P G, Verykios V S. Duplicate record detection: A survey. *IEEE Trans. Knowledge and Data Engineering*, 2007, 19(1): 1-16.

[4] Ektefa M, Jabar M A, Sidi F, Memar S, Ibrahim H, Ramli A. A threshold-based similarity measure for duplicate detection. In *Proc. IEEE Conf. Open Systems*, September 2011, pp.37-41.

[5] Gao C, Hong X G, Peng Z H, Chen H D. Web trace duplication detection based on context. In *Proc. the Int. Conf. Web Information Systems and Mining*, September 2011, pp.292-301.

[6] Das D, Martins A F T. A Survey on Automatic Text Summarization. The MIT Press, 2007.

[7] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 2003, 3: 993-1022.

[8] Landauer T K, Foltz P W, Laham D. An introduction to latent semantic analysis. *Discourse Processes*, 1998, 25(2/3): 259-284.

[9] Hofmann T. Probabilistic latent semantic analysis. In *Proc. the 15th Conf. Uncertainty in Artificial Intelligence*, August 1999, pp.289-296.

[10] Kim D, Wang H X, Oh A. Context-dependent conceptualization. In *Proc. the 23rd Int. Joint Conf. Artificial Intelligence*, August 2013, pp.2654-2661.

[11] Guo S T, Dong X L, Srivastava D, Zajac R. Record linkage with uniqueness constraints and erroneous values. *Proc. the VLDB Endowment*, 2010, 3(1/2): 417-428.

[12] Sun L W, Franklin M J, Krishnan S, Xin R S. Fine-grained partitioning for aggressive data skipping. In *Proc. the ACM SIGMOD Int. Conf. Management of Data*, June 2014, pp.1115-1126.

[13] Sarwar B, Karypis G, Konstan J, Riedl J. Item-based collaborative filtering recommendation algorithms. In *Proc. the 10th Int. Conf. World Wide Web*, May 2001, pp.285-295.

[14] Feng J H, Li G L, Wang H N, Feng J H. Incremental quality inference in crowdsourcing. In *Proc. the 19th Int. Conf. Database Systems for Advanced Applications*, April 2014, pp.453-467.

[15] Aizawa A, Oyama K. A fast linkage detection scheme for multi-source information integration. In *Proc. the Int. Workshop on Challenges on Web Information Retrieval and Integration*, April 2005, pp.30-39.

[16] Christen P. A survey of indexing techniques for scalable record linkage and deduplication. *IEEE Trans. Knowledge and Data Engineering*, 2012, 24(9): 1537-1555.

[17] Borthwick A, Goldberg A, Cheung P, Winkel A. Batch Automated Blocking and Record Matching. The US Press, 2011.

[18] Yang Q, Li Z X, Jiang J, Zhao P P, Liu G F, Liu A, Zhu J. NokeaRM: Employing non-key attributes in record matching. In *Proc. the 16th Int. Conf. Web-Age Information Management*, June 2015, pp.438-442.

[19] Villarreal S E G, Brena R F. Topic mining based on graph local clustering. In *Proc. the 10th Int. Conf. Artificial Intelligence: Advances in Soft Computing*, November 2011, pp.201-212.

[20] Dhamankar R, Lee Y, Doan A H, Halevy A, Domingos P. iMAP: Discovering complex semantic matches between database schemas. In *Proc. the ACM SIGMOD Int. Conf. Management of Data*, June 2004, pp.383-394.

[21] Weiss S M, Indurkhya N, Zhang T, Damerau F. Text Mining: Predictive Methods for Analyzing Unstructured Information. Springer, 2005.

[22] Hassell J, Aleman-Meza B, Arpinar I B. Ontology-driven automatic entity disambiguation in unstructured text. In *Proc. the 5th Int. Conf. the Semantic Web*, November 2006, pp.44-57.

[23] Zhang X, LeCun Y. Text understanding from scratch. arXiv:1502.01710, 2016. https://arxiv.org/abs/1502.01710, August 2017.

[24] Kim S J, Lee J H. Method of mining subtopics using dependency structure and anchor texts. In *Proc. the 19th Int. Conf. String Processing and Information Retrieval*, October 2012, pp.277-283.

[25] Wu M W, Zhang C D, Lan W Y, Wu Q Q. Text topic mining based on LDA and co-occurrence theory. In *Proc. the 7th Int. Conf. Computer Science & Education*, July 2012, pp.525-528.

[26] Li GL, Wang J N, Zheng Y D, Franklin M J. Crowdsourced data management: A survey. *IEEE Trans. Knowledge and Data Engineering*, 2016, 28(9): 2296-2319.

[27] Doan A H, Ramakrishnan R, Halevy A Y. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 2011, 54(4): 86-96.

[28] Gu B B, Li Z X, Yang Q, Xie Q, Liu A, Liu G F, Zheng K, Zhang X L. Web-ADARE: A web-aided data repairing system. *Neurocomputing*, 2017, 253: 201-214.

[29] Li G L, Chai C L, Fan J, Weng X P, Li J, Zheng Y D, Li Y B, Yu X, Zhang X H, Yuan H T. CDB: Optimizing queries with crowd-based selections and joins. In *Proc. the ACM Int. Conf. Management of Data*, May 2017, pp.1463-1478.

[30] Jiang L L, Wang Y F, Hoffart J, Weikum G. Crowdsourced entity markup. In *Proc. the 1st Int. Conf. Crowdsourcing the Semantic Web*, October 2013, pp.59-68.

[31] Wang J N, Kraska T, Franklin M J, Feng J H. Crowder: Crowdsourcing entity resolution. *Proc. the VLDB Endowment*, 2012, 5(11): 1483-1494.

[32] Gu B B, Li Z X, Zhang X L, Liu A, Liu G F, Zheng K, Zhao L, Zhou X F. The interaction between schema matching and record matching in data integration. *IEEE Trans. Knowledge and Data Engineering*, 2017, 29(1): 186-199.

[33] Demartini G, Difallah D E, Cudré-Mauroux P. ZenCrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proc. the 21st Int. Conf. World Wide Web*, April 2012, pp.469-478.

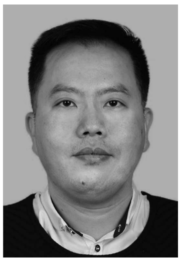
[34] Gokhale C, Das S, Doan A H, Naughton J F, Rampalli N, Shavlik J, Zhu X J. Corleone: Hands-off crowdsourcing for entity matching. In *Proc. the ACM SIGMOD Int. Conf. Management of Data*, June 2014, pp.601-612.

Zhi-Xu Li is an associate professor in the School of Computer Science and Technology at Soochow University, Suzhou. He worked as a research fellow at King Abdullah University of Science and Technology, Thuwal. He received his Ph.D. degree in computer science from the University of Queensland, Queensland, in 2013, and his B.S. and M.S. degrees in computer science from Renmin University of China, Beijing, in 2006 and 2009 respectively. His research interests include data cleaning, big data applications, information extraction and retrieval, machine learning, deep learning, knowledge graph and crowdsourcing.

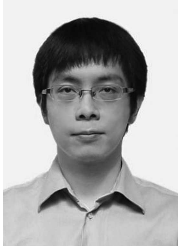
Qiang Yang is a Master student in the School of Computer Science and Technology at Soochow University, Suzhou. His research interests include data cleaning, data integration, information extraction, machine learning, crowdsourcing, and knowledge graph.

An Liu is an associate professor in the School of Computer Science and Technology at Soochow University, Suzhou. Prior to that in 2014, he was a senior research associate in the Joint Research Center of City University of Hong Kong (CityU), Hong Kong, and University of Science and Technology of China (USTC), Hefei. He received his Ph.D. degree in computer science from both CityU and USTC in 2009. His research interests include security, privacy, and trust in emerging applications and services computing.

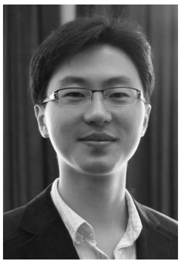
Guan-Feng Liu is an associate professor in the school of Computer Science and Technology at Soochow University, Suzhou. He received his Ph.D. degree in computer science from Macquarie University, Sydney, in 2013. His research interests include social network mining and trust. He has published over 40 papers in the most prestigious journals and conferences.



Jia Zhu is an associate professor in the School of Computer, South China Normal University, Guangzhou. He received his Bachelor's degree in information technology in 2004 and his Master's degree in information technology (Hons) in 2006, both from Bond University, Golden Coast, and his Ph.D. degree in computer science from University of Queensland, Queensland. His research interests include data mining, machine learning, and artificial intelligence.



Jia-Jie Xu is an associate professor of Soochow University, Suzhou. He is a member of the Advanced Data Analytics Research Center, Suzhou. He received his Ph.D. and Master's degrees from Swinburne University of Technology, Melbourne, and University of Queensland, Queensland, in 2006 and 2011 respectively, and then worked in the Institute of Software, Chinese Academy of Sciences, Beijing, as an assistant professor before joining Soochow University. His research interests include spatio-temporal database systems, big data analytics, and mobile computing.



Kai Zheng is a professor in the School of Computer Science and Technology at Soochow University, Suzhou. He received his Ph.D. degree in computer science from the University of Queensland, Queensland, in 2012. His research interests include finding effective and efficient solutions for managing, integrating and analyzing big data for business, scientific and personal applications. He has been working in the area of spatial-temporal databases, uncertain databases, trajectory computing, social-media analysis and bioinformatics. He has published over 60 papers in the highly referred journals and conferences such as SIGMOD, ICDE, EDBT, The VLDB Journal, ACM Transactions, and IEEE Transactions.

Min Zhang is a distinguished professor in the School of Computer Science and Technology, Soochow University, Suzhou. He received his Bachelor's degree and Ph.D. degree in computer science from Harbin Institute of Technology, Harbin, in 1991 and 1997, respectively. His current research interests include machine translation, natural language processing, and artificial intelligence.