Computing and Informatics, Vol. 38, 2019, 621–646, doi: 10.31577/cai_2019_3_621

DISCOVERING FOREIGN KEYS ON WEB TABLES WITH THE CROWD

Xiaoyu Wu, Ning Wang*, Huaxi Liu

School of Computer and Information Technology, Beijing Jiaotong University
Beijing 100044, China
e-mail: {16112087, nwang, 13120407}@bjtu.edu.cn

Abstract. Foreign-key relationship is one of the most important constraints between two tables. Previous works focused on detecting inclusion dependencies (INDs) or foreign keys in relational database. To discover foreign-key relationship is obviously helpful for analyzing and integrating data in web tables. However, because of poor quality of web tables, it is difficult to discover foreign keys by existing techniques based on checking basic integrity constraints. In this paper, we propose a hybrid human-machine framework to detect foreign keys on web tables. After discovering candidates and evaluating their confidence of being true foreign keys by machine algorithm, we verify those candidates leveraging the power of the crowd. To reduce the monetary cost, a dynamical task selection technique based on conflict detection and inclusion dependency is proposed, which could eliminate redundant tasks and assign the most valuable tasks to workers. Additionally, to make workers complete tasks more effectively and efficiently, sampling strategy is applied to minimize the number of tuples posed to the crowd. We conducted extensive experiments on real-world datasets and results show that our framework can obviously improve foreign key detection accuracy on web tables with lower monetary cost and time cost.

Keywords: Foreign key, web tables, crowdsourcing, task selection, task reduction, semantic recovery

* Corresponding author

1 INTRODUCTION

Foreign-key relationship is one of the most important constraints between two tables. Previous works focus on detecting inclusion dependencies (abbr. INDs) or foreign keys in relational database [1, 2, 3, 4], in which table name, uniqueness of key and strict inclusion dependency between foreign key (the dependent attribute) and primary key (the referenced attribute) are important factors for detecting foreign keys. The worldwide web contains a vast amount of tables on varieties of topics [5], and to discover foreign-key relationship is obviously helpful for analyzing and integrating data in web tables. Unfortunately, all the previous works could not be used directly on web tables which often lose table names and sometimes have noisy data. In fact, web tables may not satisfy the entity integrity constraint and referential integrity constraint. Figure 1 shows fragments of typical web tables from Google Table [6] which miss table names and have duplicated tuples. There is a foreign key relationship between tables in Figure 1, where country in Figure 1 b) is a foreign key referencing short name in Figure 1 a). However, foreign key detection method in relational database could not be used in such tables which lose some schema information and also do not satisfy basic integrity constraints. Furthermore, even if a web table has a table name, it often does not include meaningful information which could describe the semantics of this table exactly. Because of poor quality of web tables, it is difficult to discover foreign keys effectively only by machine algorithm.

Short name	Country code	Currency code
United Kingdom	UK	GBP
United States	US	USD
Russia	RU	RUB
South Korea	KR	KRW
Iraq	IR	IRR
Iraq	IQ	IQD
China	CN	CNY
Sudan	SD	SDG
Poland	PL	PLN
Namibia	NA	NAD
Zambia	ZM	ZMW

a)

Company Name	Country	State/Province
1-800-Balloons.com	United States	Nevada
10 Minutes With	United Kingdom	
118Boardshop.com	United States	California
11am.co.kr	South Korea	
121doc.net	UK	
17ugo.com	China	

b)

Figure 1. An example of web tables

Recent researches have shown that crowdsourcing could be used effectively to solve problems that are difficult for computers, such as entity resolution [7], sentiment analysis [8], and image recognition [9]. We propose a hybrid human-machine framework that leverages human intelligence to discover foreign keys on web tables effectively. Our framework implements foreign key detection in two phases, which are finding candidates by machine algorithm and validating candidates by the crowd.

The first phase is for candidate generation. Because web tables may not satisfy the entity integrity constraint and referential integrity constraint, we define uniqueness degree to measure the proportion of unique values in a column and coverage rate to measure the proportion of dependent attribute values contained in the referenced attribute. Then we use four features to evaluate the possibility of a candidate being a true foreign key which are the unique degree of the referenced attribute, the coverage rate of the referenced attribute to the dependent attribute, the column names' similarity and whether the dependent attribute is a key. Short of semantics, it is so difficult for computer to understand relationships between two tables that will result in some false positive candidates. Fortunately, with the intelligence of humans, those false positive candidates can be easily distinguished.

After the first phase, candidates are generated inevitably with some false positives, so crowdsourcing is used for distinguishing true foreign keys from all candidates. As the crowd is not free, cost control is one of the biggest challenges in data management with the crowd [10]. To reduce the monetary cost, number of tasks should be reduced. Considering some conflicts in candidates and inclusion dependency between dependent attributes, we propose dynamical task selection methods based on conflict detection and inclusion dependency. The experimental results show our method can effectively reduce the number of tasks.

Besides the monetary cost, time cost is also to be considered for crowdsourcing tasks. For foreign key validation, workers have to check content between two tables. Facing tables with too many tuples, workers will be impatient with taking long time to browse the whole table and make decision. So, to reduce the latency of tasks, we propose a task reduction method based on sampling strategy, which could reduce the volume of web tables under the condition that the original relationship between tables could be held.

The main contributions of this paper are:

- To our best knowledge, we are the first to propose a hybrid human-machine framework for discovering foreign keys on web tables which may not satisfy the entity integrity constraint and referential integrity constraint.
- To reduce monetary cost for crowdsourcing tasks, we propose a dynamical task selection technique based on conflict detection and inclusion dependency, which could eliminate redundant tasks and assign the most valuable tasks to workers.
- To avoid latency of tasks, we propose a task reduction method based on sampling strategy to minimize the number of tuples posed to the crowd.
- Based on real-world datasets, we evaluated the performance of human-machine hybrid approach and effectiveness of our dynamical task selection method and task reduction method.

The remainder of the paper is organized as follows. We present solution overview in Section 2. Section 3 gives the machine algorithm for generating and scoring foreign key candidates on web tables. Our dynamical task selection method and task reduction method are discussed in Section 4 and Section 5, respectively. Then

we report results of experiment in Section 6, discuss related works in Section 7, and conclude the paper in Section 8.

2 SOLUTION OVERVIEW

To discover foreign keys on web tables, we propose a hybrid human-machine framework. Our framework takes as input a set of web tables and generates foreign key candidates by machine algorithm. Then the false positive candidates are verified by the crowd and true foreign keys are output. Figure 2 shows the framework.

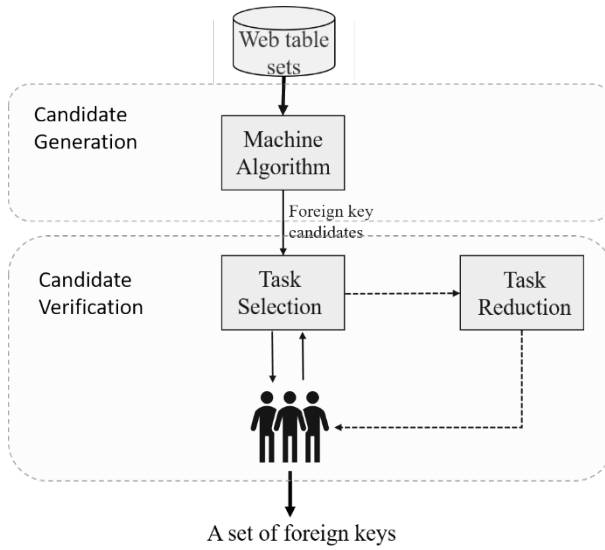


Figure 2. A hybrid human-machine framework for discovering foreign keys on web tables

There are two phases in our framework. In the first phase, machine algorithm is used to find candidates on input tables and calculate their confidences to be true foreign keys (the details will be described in Section 3). Then candidates with high confidence will be verified by the crowd in the second phase. As the crowd is not free, we take measures to reduce monetary cost by reducing the number of tasks. Because there are some tasks that could be deduced by other tasks, we dynamically select tasks based on conflict detection (Section 4.1) and inclusion dependency (Section 4.2). The task selection method based on conflict detection reduces tasks in conflict with those verified as true, while the task selection method based on inclusion dependency reduces tasks which can be deduced by those verified with the method. Browsing a whole table with a large volume will surely make workers impatient and lead a high latency. So, we try to reduce tables' volume leveraging a combinational sampling strategy (Section 5). After steps above, only most valu-

able tasks are posted to the crowd through the crowdsourcing platform, and the final verified results are returned.

3 FOREIGN KEY CANDIDATE GENERATION

In our framework for discovering foreign keys on web tables, generating candidates is the first step.

Definition 1 (Uniqueness Degree)**.** Given table S and its attribute $S.b$, the uniqueness degree of $S.b$, denoted as $UNI(S.b)$, is the ratio of $S.b$'s cardinality to S's cardinality.

$$UNI(S.b) = \frac{|S.b|}{|S|}. \tag{1}$$

Definition 2 (Coverage Rate)**.** Given a pair of tables R, S and attributes $R.a$, $S.b$ in R and S, respectively, the coverage rate of $S.b$ on $R.a$, denoted as $COV(S.b, R.a)$, could be calculated using following formula:

$$COV\,(S.b, R.a) = \frac{|R.a \cap S.b|}{|R.a|}. \tag{2}$$

We start the detection from measuring the confidence of attribute pairs to be true foreign keys. As web tables may lose or duplicate some cells or tuples, we relax checking the uniqueness of key and containment relationship between key and foreign key which are necessary conditions for foreign key detection in relational database. Let δ be the threshold of primary keys' uniqueness degree, λ be the threshold of primary keys' coverage rate to the foreign key, and $p = (R.a, S.b)$ denote a pair of attributes where R and S are corresponding tables. For attribute pair $(R.a, S.b)$ with $UNI(S.b) \geq \delta$ and $COV(S.b, R.a) \geq \lambda$, we use a scoring function $CTF(R.a, S.b)$ to measure the attribute pair's confidence to be a true foreign key. The scoring function is a weighted sum of 4 scores corresponding to 4 features as follows:

S.b's unique degree: $Score_1 = UNI(S.b)$.

S.b's coverage to R.a: $Score_2 = COV(S.b, R.a)$.

The similarity between attribute name of R.a and S.b: $Score_3 = Sim(R.a, S.b)$.

Whether R.a is a key: $Score_4 = 1$ if R.a is a key ($UNI(R.a) \geq \delta$) otherwise $Score_4 = 0$.

$$CTF\,(R.a, S.b) = \sum_{i=1}^{4} \omega_i Score_i. \tag{3}$$

In Equation (3), $0 < \omega_1, \omega_2, \omega_3 < 1, \omega_4 < 0$. If $CTF(R.a, S.b)$ is higher than the threshold of confidence, $(R.a, S.b)$ is recognized as a foreign key candidate and denoted as $R.a \xrightarrow{\delta,\lambda} S.b$.

An important problem in this step is how to evaluate the string similarity. Generally, the similarity matching algorithm could be accurate matching and fuzzy matching. Accurate matching is usually used in traditional inclusion dependencies discovery in relational database [1, 2, 4]. For web tables often with noisy data, we use edit distance to evaluate the similarity between attributes' values and Jaro Winkler Distance [11] to measure similarity between attributes' names.

Though we try our best to improve accuracy of the candidate generation method, there are still many false positive candidates in the result. Therefore, we decide to utilize human intelligence to find true foreign keys from candidates.

4 DYNAMICAL TASK SELECTION

For discovering foreign keys on web tables, we adopt a human-machine hybrid approach which first uses machine algorithm to generate a foreign key candidate set, and then ask humans to verify candidates in the set as either foreign-key or non-foreign-key. As the crowd is not free, cost control is one of the most important problems in crowdsourced data management, and appropriate task selection will surely make the crowd work more efficiently. For reducing monetary cost, we dynamically detect redundant tasks and assign the most valuable tasks to workers. In this section, we propose the dynamical task selection method based on conflict detection and inclusion dependency between dependent attributes.

Since, in our setting, some candidates will be verified by crowd, and others will be deduced by the task selection method. We call the former as *crowdsourced*(*labeled*) *candidates*, and the latter as *deduced*(*labeled*)*candidates*.

4.1 Task Selection Based on Conflict Detection

In a list of foreign key candidates, there often exist some conflicts. If a candidate is verified to be true, its conflicts must be false. We could utilize conflict relationship between candidates to reduce number of crowdsourcing tasks.

Definition 3 (Foreign Key Candidates Reference Graph)**.** Given a set of foreign key candidates FC, a foreign key candidates reference graph is a weighted directed graph $FKRG = \langle \zeta, \varphi, \omega \rangle$, where:

- ζ is a set of attributes occurred in FC.
- φ is a set of foreign key candidates, and $\langle R.a, S.b \rangle \in \varphi$ iff $R.a \xrightarrow{\delta,\lambda} S.b \in FC$.
- ω is a set of weights, each of which corresponds to confidence on a foreign key candidate in φ.

Figure 3 is an example of foreign key candidates reference graph. Given two attributes $A.a$ and $B.b$, if $A.a$ is a candidate foreign key referencing $B.b$, there will be a directed edge from $A.a$ to $B.b$, this edge's weight (i.e. 0.53) represents the confidence of the foreign key candidate $R.a \xrightarrow{\delta,\lambda} S.b$.

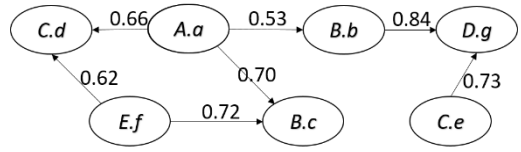


Figure 3. Foreign key candidates reference graph

A foreign key candidates reference graph is made up of vertexes and edges. While edges have weights with them reflecting the confidence of candidates, vertexes should be given weights reflecting average confidence of candidates related. The weight of a vertex is defined in Definition 4 as its influence.

Definition 4 (Influence of Attributes)**.** Given a foreign key candidates reference graph $FKRG = \langle \zeta, \varphi, \omega \rangle$, $T.m \in \zeta$, $RS(T.m) = \left\{ R.a \xrightarrow{\delta,\lambda} S.b | R.a = T.m \text{ or } S.b = T.m \right\}$, the influence of $T.m$, denoted as $\mathit{Influence}(T.m)$, could be calculated by the following formula:

$$\mathit{Influence}\,(T.m) = \frac{\sum_{i=1}^{|RS(T.m)|} CTF\,(R.a, S.b)}{|RS\,(T.m)|} \tag{4}$$

where $R.a \xrightarrow{\delta,\lambda} S.b \in RS(T.m)$.

For example, in Figure 3, $\mathit{Influence}\,(A.a) = \frac{CTF(A.a,B.b)+CTF(A.a,B.c)+CTF(A.a,C.d)}{3} = \frac{0.66+0.53+0.70}{3} = 0.63$. Intuitively, if a vertex has high influence, candidates related to this vertex may have high confidence of being true foreign keys and are more likely to be verified as a true foreign key.

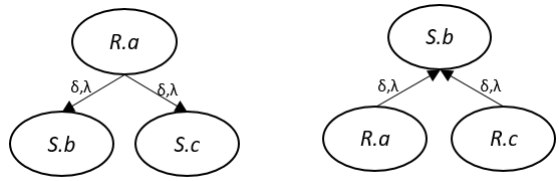
In a true foreign key relationship, the dependent attribute couldn't be contained in the set of values of many other attributes, while the referenced attribute couldn't be referenced by multiple attributes from one table. Combining with web tables' characteristics, we get conflict detection rules below.

Conflict Detection Rules:

- A foreign key can only reference one primary key in the same referenced table.
- A primary key can only be referenced by one foreign key in the same dependent table.

Figure 4 gives examples about conflict rules. In Figure 4 a), there are foreign key candidates $R.a \xrightarrow{\delta,\lambda} S.b$ and $R.a \xrightarrow{\delta,\lambda} S.c$. In case any candidate is verified to be a true foreign key, another will be ruled out. This case indicates the similarity of b and c is very high, and there exists data redundancy in S. In Figure 4 b), there are foreign key candidates $R.a \xrightarrow{\delta,\lambda} S.b$ and $R.c \xrightarrow{\delta,\lambda} S.b$. In the same way, when any

candidate is verified to be a true foreign key, another will be removed. This case indicates the similarity of attribute $R.a$ and $R.c$ is very high, and there exist data redundancy in R.



a) b)

Figure 4. Conflicts of foreign key candidates

Based on the conflict detection rule, we propose to deduce unlabeled candidates with the labeled ones. Details of this candidate deduction algorithm are shown in Algorithm 1.

Algorithm 1 Candidate Deduction Based on Conflict Detection

Input: LC: a set of candidates that have been labeled as true foreign keys;
uc: an unlabeled candidate;
Output: rc: the deduced result of uc;

```
1: begin
2: rc ← null;
3: for ∀c ∈ LC do
4:    Conf ← false;
5:    F ← AttrSame(c.ref, uc.ref);
6:    if F then
7:       Conf ← TableSame(c.dep.t, uc.dep.t);
8:    else
9:       F ← AttrSame(c.dep, uc.dep);
10:      if F then
11:         Conf ← TableSame(c.ref.t, uc.ref.t);
12:      end if
13:   end if
14:   if Conf then
15:      rc ← lable(uc, false);
16:   end if
17: end for
18: return rc;
19: end
```

Given a set of candidates LC that have been labeled as true foreign keys and an unlabeled candidate uc, for each labeled candidate in LC, this algorithm check

whether its referenced attribute is the same with the referenced attribute of unlabeled candidate (lines 3–5). If it is the same, check whether their dependent attributes are from the same table, and label the unlabeled candidate as a conflict if the checking result is true (lines 6–7). Otherwise, check whether its dependent attribute is the same with the dependent attribute of unlabeled candidate or not, if it is the same, check whether their referenced attributes are from the same table, and label the unlabeled candidate as a conflict if the checking result is true (lines 8–13). The unlabeled candidate will be deduced as a non-foreign-key if it is labeled as conflict candidate (lines 14–16). Finally, the deduced result will be returned (lines 18–19).

Task selection aims at reducing crowdsourcing cost by reducing the number of tasks (i.e. crowdsourced candidates). Using the candidate deduction algorithm based on conflict detection, we dynamically reduce tasks (i.e. candidates) which can be deduced and select the most valuable task to be crowdsourced. When publishing tasks to crowdsourcing platform, we should give priority to the candidate which is most likely to be a true foreign key. Once a true foreign key is confirmed, other candidates conflicting with it will be removed from the task list.

The dynamical task selection method based on conflict selects tasks from two perspectives. From the global perspective, it chooses the high-influence vertex first. Then it chooses the foreign key candidates with high confidence first from the local perspective. Details of this method are shown in Algorithm 2.

Algorithm 2 Dynamical Task Selection Method Based on Conflict

Input: G: a foreign key candidates reference graph;
Output: R: the labeled result of candidates in G;

1: **begin**
2: $flag \leftarrow hasEdge(G)$;
3: **while** *flag* **do**
4: $v_h \leftarrow getTopV(G)$;
5: $CS \leftarrow asDepAttr(v_h)$;
6: $cfk_h \leftarrow getTopE(CS)$;
7: $r \leftarrow crowd(cfk_h), R \leftarrow crowd(cfk_h)$;
8: $elimEdge(cfk_h, G)$;
9: **if** r **then**
10: $R \leftarrow deduceCan(r, CS), cc \leftarrow confCan(r, CS)$;
11: $elimEdge(cc, G)$;
12: **end if**
13: **end while**
14: **return** R;
15: **end**

Given a foreign key candidates reference graph G, we first calculate each vertex's weight and select the one (denoted as v_h) with the highest weight (lines 2–4). Then from the foreign key candidate set where v_h is a dependent attribute, we choose the

candidate cfk_h with the highest confidence to be verified by the crowd (lines 5–7). After getting the verification result, we eliminate the corresponding edge of cfk_h from G (line 8). If cfk_h is verified to be a true foreign key, unlabeled candidates will be deduced using Algorithm 1, and corresponding edges will be removed (lines 9–11). Above steps are repeated until there is no edge in G, and then the labeled result of candidates in G will be returned (lines 13–15).

4.2 Task Selection Based on Inclusion Dependency

In addition to conflicts, there may be an inclusion dependency between dependent attributes which refer to the same referenced attribute. In this section, we introduce another task selection method based on inclusion dependency.

Suppose there are three foreign key candidates $A.a \xrightarrow{\delta,\lambda} B.b$, $C.c \xrightarrow{\delta,\lambda} B.b$ and $A.a \xrightarrow{\delta,\lambda} D.d$ that have been verified to be true foreign keys, and $COV(B.b, A.a) = COV(B.b, C.c) = COV(D.d, A.a) = 1$, i.e. $A.a \subseteq B.b$, $C.c \subseteq B.b$, $A.a \subseteq D.d$. If $C.c \subseteq A.a$, it is easy to get $C.c \subseteq D.d$. Because foreign key is a semantic relationship between attributes [1], we can get the conclusion that

1. the semantics of $B.b$ is similar to the semantics of $A.a$ and $C.c$, and
2. the semantics of $A.a$ is similar to the semantics of $D.d$.

Thus, we infer that

1. $A.a$ and $C.c$ are semantically related, and
2. $C.c$ and $D.d$ are semantically related.

Based on the above conditions, the candidate $C.c \xrightarrow{\delta,\lambda} D.d$ could be deduced as a true foreign key. This discovery gives us an inspiration of deducing candidates based on inclusion dependency between dependent attributes which is the core of this task selection method (based on inclusion dependency).

Next, we will describe the candidate deduction method based on inclusion dependency. Algorithm 3 gives the details of the method.

Given a set of candidates LC that have been labeled as true foreign keys and an unlabeled candidate uc, denote the referenced attribute and dependent attribute of uc as $P.k$ and $F.k'$, respectively (line 2), the algorithm first check whether there is any labeled candidate (denoted as c) of which the dependent attribute is the same with $F.k$ and coverage equals to 1 (lines 3–6). If any, it will try to discover another two labeled candidates (denoted as c' and c'') which have the same dependent attribute and satisfy

1. referenced attribute of c' is the same with referenced attribute of c and referenced attribute of c'' is the same with $P.k$,
2. the coverage of c' and c'' are equal to 1,
3. all values in $F.k'$ are a subset of dependent attribute of c'.

Algorithm 3 Candidate Deduction Based on Inclusion Dependency

Input: LC: a set of candidates that have been labeled as true foreign keys;
uc: an unlabeled candidate;
Output: rc: the deduced result of uc;

```
1: begin
2: rc ← null, P.k ← uc.refAttrrc, F.k′ ← uc.depAttr;
3: for ∀c ∈ LC do
4:    if c.depAttr == F.k && c.coverage == 1 then
5:       C.r ← c.refAttr, LC′ ← LC − {c};
6:    end if
7:    S ← Pair2Can(LC′);
8:    while S do
9:       s ← getEle(S), c′ ← s.c1, c′ ← s.c2, remove(s);
10:      if c′.depAttr == c″.depAttr && c′.refAttr == C.r && c′.refAttr ==
         P.k && c′.coverage == c″.coverage == 1 && Fk′ ⊆ c′.depAttr then
11:         rc ← label(uc, true);
12:         break;
13:      end if
14:   end while
15:   if rc then
16:      break;
17:   end if
18: end for
19: return rc;
20: end
```

If these candidates exist, then uc will be deduced as a true foreign key (lines 7–14). Finally, the deduced result will be returned (lines 15–19).

Example 1. For an unlabeled candidate $C.c \xrightarrow{\delta,\lambda} D.d$ with $COV(D.d, C.c) = 1$ (i.e. $C.c \subseteq D.d$), we check whether there is any candidate $C.c \xrightarrow{\delta,\lambda} B.b$ with $COV(B.b, C.c) = 1$ has been labeled as true. If any, we try to detect the labeled candidates which have the same dependent attribute ($T.m$) and satisfy the following conditions:

1. $T.m$ reference to $B.b$ and $D.d$ at the same time,
2. $COV(B.b, T.m) = COV(D.d, T.m) = 1$,
3. $C.c \subseteq D.d$.

If these candidates exist, then the candidate $C.c \xrightarrow{\delta,\lambda} D.d$ can be deduced as a true foreign key.

Generally, candidates deduced with high confidence are more credible. So, we are inclined to let candidates with low confidence be crowdsourced and candidates

with high confidence to be deduced. Given a set of candidates to be verified, the task selection method based on inclusion dependency sorts them by their confidence in increasing order and gives priority to candidates with higher confidence first. Each time, candidate with the lowest confidence will be checked whether it can be deduced based on inclusion dependency. If not, its corresponding task will be posted to the crowdsourcing platform, otherwise, it will be deduced as a true foreign key.

4.3 A Combined Task Selection Method

To reduce the number of tasks as much as possible, we combine the task selection method based on conflict detection and inclusion dependency in practice. The above task selection methods (in Sections 4.1 and 4.2) publish the most valuable candidate to the crowdsourcing platform. Hence, long latency would be caused when only one candidate is posted to crowd at a time. To overcome this drawback, the combined task selection method places multiple candidates into a single task.

The core of the combined method is detecting redundant candidates in the task. There are two kinds of redundant candidates, certainly redundant candidates which could be deduced by labeled candidates and probably redundant candidates which are probably deduced by other unlabeled candidates in the same task.

Definition 5 (Certainly redundant candidate)**.** Given a set of labeled candidates C and an unlabeled candidate uc which will be crowdsourced, if uc could be deduced by candidates in C with any candidate deduction method (i.e. candidate deduction based on conflict detection or inclusion dependency), we say uc is a certainly redundant candidate.

Definition 6 (Probably redundant candidates)**.** Let UC be a set of unlabeled candidates which will be crowdsourced together in a single task, and uc is one of them, suppose all candidates except uc will be labeled as true foreign keys by crowd. Let C' be the labeled result set. If uc could be deduced by any candidates in C' with any candidate reduction method, we say uc is a probably redundant candidate in UC.

Next, we will introduce how to check and deal with two kinds of redundant candidates in a single task. Suppose k unlabeled candidates will be placed into a single task each time. For each unlabeled candidate, we need to check whether it could be deduced by labeled candidates first. If any, it will be recognized as a certainly redundant candidate and be deduced. A certainly redundant candidate will be removed and never be crowdsourced. Then the unlabeled candidates left should be checked whether they are probably redundant candidates in the task. For each candidate left, we suppose all the others will be labeled as true foreign keys and check whether it could be deduced with any candidate deduction method. If not, it will be recognized as a valuable candidate, otherwise, it will be recognized as a probably redundant candidate. Only valuable candidates will be placed into the crowdsourcing task, and probably redundant candidates will be hold on. After

crowdsourcing results coming up, some probably redundant candidates might be deduced, and others not be deduced will be added into the unlabeled candidate set again. Algorithm 4 shows the details of the redundant candidate detection method.

Algorithm 4 Redundant Candidates Detection

Input: LC: a set of candidates which have been labeled as true foreign keys;
ULC: a set of candidates that need to be crowdsourced;
Output: RLC: a candidate set without redundant candidate

```
1: begin
2: if ∃c ∈ LC then
3:    ULC ← DeduceConflict(LC, ULC);
4:    ULC ← DeduceIND(LC, ULC);
5: end if
6: Temp ← Label(ULC, true);
7: ULC ← DeduceConflict(Temp, ULC);
8: RLC ← DeduceIND(Temp, ULC);
9: return RLC;
10: end
```

Given a set of candidates that need to be crowdsourced, we first deduce certainly redundant candidates with the candidate deduction methods (lines 2–5). Then we suppose all candidates left will be labeled as true foreign keys (line 6) and detect probably redundant candidates, i.e. detect whether they could be deduced by other candidates when they are labeled as true foreign keys (lines 7–8). Here, the procedure DeduceConflict is used to deduce candidates based on conflict detection, and the procedure DeduceIND is used to deduce candidates based on inclusion dependencies. Finally, the candidate set without redundant candidate will be returned (line 9).

Algorithm 5 gives the details of the combined task selection method.

Given a set of unlabeled foreign key candidates, we first sort them by their confidence in increasing order and select the top-k candidates to make up a task (lines 2–5). Then we remove the redundant candidates in the task and check the number of tasks (lines 8–9). If the number is less than k, we update the tasks (i.e. add new unlabeled tasks into the task) and repeat steps above until the number of candidates in the task is not less than k (lines 10–17).

Example 2. Consider the foreign key candidates in Figure 3. Suppose 5 candidates are contained in a task, $COV(C.d, A.a) = COV(C.d, E.f) = COV(B.c, A.a) = 1$ and $E.f \subseteq A.a$. Firstly, we sort these candidates by their confidence in increasing order and get the top-5 candidates, i.e., $A.a \xrightarrow{\delta,\lambda} B.b$, $E.f \xrightarrow{\delta,\lambda} C.d$, $A.a \xrightarrow{\delta,\lambda} C.d$, $A.a \xrightarrow{\delta,\lambda} B.c$ and $E.f \xrightarrow{\delta,\lambda} B.c$. For there is no labeled candidate, we just detect probably redundant candidate. For each candidate, suppose all the others will be labeled as true foreign keys. $A.a \xrightarrow{\delta,\lambda} B.c$ and $A.a \xrightarrow{\delta,\lambda} B.b$ are conflict candidates. In this case, we should crowdsource candidates with high confidence first. So, $A.a \xrightarrow{\delta,\lambda}$

Algorithm 5 The Combined Task Selection Method

Input: LC: a set of candidates that have been labeled as true foreign keys;
ULC: an unlabeled candidate set;
k: the number of candidates that need to be placed into a task;
Output: T: the task that needs to be crowdsourced;

```
1: begin
2: if ∃c ∈ ULC then
3:    S ← SortInc(ULC);
4:    T ← getTopK(S);
5:    RD ← true;
6:    while RD do
7:       T ← RedundancyDetect(T, LC);
8:       if Num(T) < k then
9:          RD ← true;
10:         if RD then
11:            T ← Update(T);
12:         end if
13:      else
14:         RD ← false;
15:      end if
16:   end while
17: end if
18: return T;
19: end
```

$B.c$ will be recognized as a probably redundant candidate. In addition, $E.f \xrightarrow{\delta,\lambda} B.c$ might be deduced by $A.a \xrightarrow{\delta,\lambda} B.c$, $E.f \xrightarrow{\delta,\lambda} C.d$ and $A.a \xrightarrow{\delta,\lambda} C.d$, therefore, it will be recognized as a probably redundant candidate and be temporarily removed. After eliminating redundant candidates, there are only 3 valuable candidates left. So, we need add another two tasks (i.e. $C.e \xrightarrow{\delta,\lambda} D.g$ and $B.b \xrightarrow{\delta,\lambda} D.g$) into the task and check the redundant tasks again. If $A.a \xrightarrow{\delta,\lambda} B.c$ is verified to be a true foreign key by crowd, $A.a \xrightarrow{\delta,\lambda} B.b$ will be labeled as false directly, otherwise, it will be added into subsequent task to be verified by crowd. Likewise, if $A.a \xrightarrow{\delta,\lambda} B.c$, $E.f \xrightarrow{\delta,\lambda} C.d$ and $A.a \xrightarrow{\delta,\lambda} C.d$ are verified to be true foreign keys, $E.f \xrightarrow{\delta,\lambda} B.c$ will be deduced as a true foreign key, otherwise, it will be added into subsequent task group.

5 TASK REDUCTION WITH SAMPLING

Browsing a large volume table will surely make workers impatient and lead a high cost. To control the tasks' latency, we propose a sampling strategy to sample some representative tuples to prompt to workers.

Generally, the sampling method can be divided into a uniform sampling and a biased sampling. A foreign key relationship involves two tables. Although records in single table are independent, there exists dependency among the attribute pair in a foreign key relationship. Hence, uniform sampling could not be used simply in this situation. We propose a combinational sampling method which can keep the original relationship between tables while reducing the tables' volume.

Our task reduction method based on sampling strategy consists of two phases: dependent table reduction and referenced table reduction. Suppose there is a candidate foreign key relationship $R.a \xrightarrow{\delta,\lambda} S.b$ between table R and S. We sample the dependent table R with the uniform sampling and get the reduced table R'. Based on R', we sample the referenced table S with a biased sampling method. We describe the details as follows.

1. Dependent table reduction: Suppose there are m tuples in the dependent table R, the sample rate is ϕ. Considering that each tuple in the table has the same probability to be sampled though some of them may have the same value on dependent attribute, we randomly sample $\lfloor m \times \phi \rfloor$ tuples from R to make up the reduced dependent table R'.
2. Referenced table reduction: Suppose there are n tuples in the referenced table S, and the uniqueness degree of the referenced attribute $S.b$ is δ', only $\lfloor n \times \phi \rfloor$ rows are allowed to be sampled. If we randomly sample the referenced table, the original relationship between two tables (the dependent table and the referenced table) could not be hold. We partition S into covered part and uncovered part. The covered part consists of tuples in which the referenced attribute is referenced by the values sampled in the dependent attribute. The uncovered part consists of tuples left when removing the covered part from S. From the covered part, we extract all the tuples in which values of the referenced attribute is referenced by values of the dependent attribute in R' (reduced dependent table). Suppose the number of tuples sampled from the covered part is k, we then randomly select $\lfloor m \times \phi \rfloor - k$ tuples from the uncovered part, and combine all the tuples we sampled into the reduced referenced table S'.

Example 3. Consider two tables in Figure 5, in which Figure 5 a) is the referenced table S with 16 tuples, and Figure 5 b) is the dependent table R with 12 tuples. There exists a foreign key relationship $R.a \xrightarrow{\delta,\lambda} S.b$ on which the uniqueness degree on $S.b$ is 0.94, and $S.b$'s coverage to $R.a$ is 0.92. If the sampling rate is 0.5, R should be reduced to 6 tuples while S should be reduced to 8 tuples. According to our sampling method, we randomly sample 6 tuples from R and make up the reduced dependent table R' (see Figure 5 d)). After getting R', we should make corresponding reduction to the referenced table. We first partition S into covered part and uncovered part, then extract 4 tuples in which values of the referenced attribute short name is referenced by values of the dependent attribute country in R' (the value of short name is United States, United Kingdom, Japan and Vietnam) from the covered

Short name	Country code	Currency code
United Kingdom	UK	GBP
United States	US	USD
China	CN	CNY
Japan	JP	JPY
Russia	RU	RUB
South Korea	KR	KRW
Sweden	SE	SEK
Turkey	TR	TRY
Iraq	IR	IRR
Iraq	IQ	IQD
Vietnam	VN	VND
Samoa	WS	WST
Sudan	SD	SDG
Poland	PL	PLN
Namibia	NA	NAD
Zambia	ZM	ZMW

a)

Company Name	Country	State/Province
1-800-Balloons.com	United States	Nevada
10 Minutes With	United Kingdom	
1000 Markets, Inc	United States	Washington
10sec Inc.	Japan	
11am.co.kr	South Korea	
121doc.net	UK	
123Mua	Vietnam	
123RF.com	United States	Illinois
123stores.com	United States	New York
139Shop.com	China	
13:E Protein Import Aktiebolag	Sweden	
17ugo.com	China	

b)

Short name	Country code	Currency code
United Kingdom	UK	GBP
United States	US	USD
China	CN	CNY
Japan	JP	JPY
Iraq	IR	IRR
Vietnam	VN	VND
Sudan	SD	SDG
Zambia	ZM	ZMW

c)

Company Name	Country	State/Province
1-800-Balloons.com	United States	Nevada
10 Minutes With	United Kingdom	
10sec Inc.	Japan	
121doc.net	UK	
123Mua	Vietnam	
123RF.com	United States	Illinois

d)

Figure 5. An example of task reduction by sampling

part, and randomly select 4 tuples from the uncovered part. The reduced referenced table S' is shown in Figure 5 c).

From this example, we can see that only small fraction of tuples in web tables can ensure the quality of crowdsourcing task. In Section 6, we will make suggestion for optimal sampling rate by comparing the performance of different sampled tasks.

6 EXPERIMENT

We evaluate our method using a number of real word web tables. The goals of our experiment are:

1. compare the performance of our candidate foreign key generation algorithm with the fast foreign key detection method proposed in [2],

2. evaluate the power of the crowd,
3. evaluate the effectiveness of the task selection method,
4. evaluate the effectiveness of the task reduction method with sampling.

Dataset: We crawled more than 1 000 web tables from Google tables [6] and selected 118 tables which have semantic relationships with each other to conduct our experiment. The content of these tables refers to sports, economy, technology, movies and so on. Tuples in these tables add up to 12 717, and the total columns are 699. For there is no declared semantic relationship between these web tables, we manually labeled 550 foreign key constraints as the "ground truth" with the help of machine algorithm.

6.1 Performance Comparison of Machine Algorithms

Before crowdsourcing, we preprocess the web tables using machine algorithms to find foreign key candidates with high probability. In this section, we compare the precision, recall and F-Measure of our candidate foreign key detection method denoted as WFD with the fast foreign key detection method denoted as FFD [2].

We run two algorithms to process the web tables, respectively and compare their performance. Under different thresholds of candidate confidence (varying from 0.5 to 1), we compare the precision, recall and the F-measure of FFD and WFD. Precision and recall are calculated by Equations (5) and (6), respectively.

$$Precision = \frac{|TAF|}{|AF|}, \tag{5}$$

$$Recall = \frac{|TAF|}{|WF|}. \tag{6}$$

Where TAF is the set of true foreign keys the machine-based algorithm discovered, AF denotes the foreign key candidates the machine-based algorithm detected, and WF is the true foreign keys in the dataset. *F-measure* is defined as the harmonic mean of precision and recall in following formula:

$$F = \frac{(1+\alpha) \times Precision \times Recall}{\alpha \times Precision + Recall} \tag{7}$$

where α is set to 1 in the experiment.

Table 1 shows the number of candidate foreign keys (FFD disc, WFD disc), true foreign keys they discovered (FFD true, WFD true). WFD finds more true foreign keys than FFD. The higher the threshold is, the less candidates they will find.

Figures 6, 7, 8 describe the precision, recall, and F-measures of the two algorithms, respectively.

When confidence threshold varies from 0.5 to 1, all precision values of WFD are higher than that of FFD. When the threshold is set to 0.9, the precision of WFD

θ	FFD disc	FFD true	WFD disc	WFD true
0.50	772	229	819	448
0.55	598	217	756	437
0.60	487	212	703	428
0.65	435	201	678	420
0.70	320	165	593	409
0.75	253	154	406	310
0.80	205	128	263	207
0.85	133	93	198	156
0.90	94	67	169	134
0.95	45	33	102	79
1.00	17	12	22	17

Table 1. Experiment result of FFD and WFD

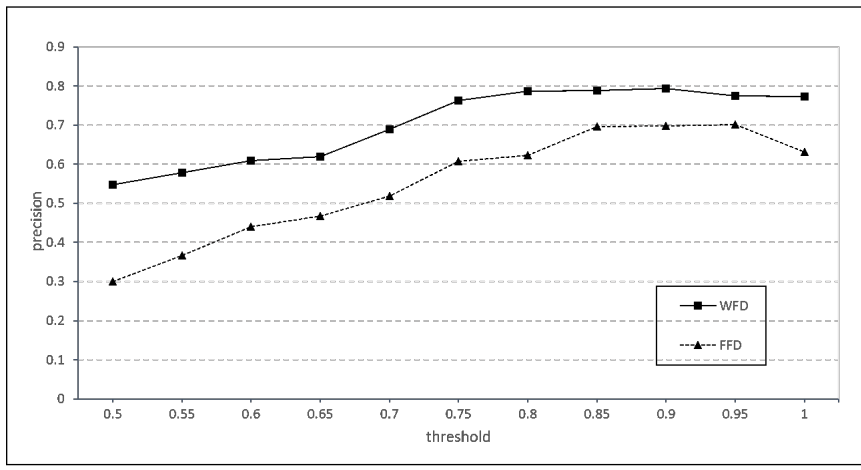


Figure 6. Precision of FFD and WFD

reaches to 79.29 %. The highest precision of FFD is 70.21 % when the threshold is 0.95. For a foreign key candidate $R.a \xrightarrow{\delta,\lambda} S.b$, FFD measures its confidence by four factors including similarity between table name of $R.a$ and $S.b$. However, most of the web tables' names could not describe the semantics of tables exactly. Eliminating the influence of web table's name makes WFD performs better than FFD in precision. The higher the confidence is, the higher the quality of foreign key candidates. Therefore, the precision of WFD and FFD trends to increase generally as confidence threshold increases.

With the increase of confidence threshold, the recall of the two methods decreases. When the threshold is less than 0.7, WFD's recall is much higher than

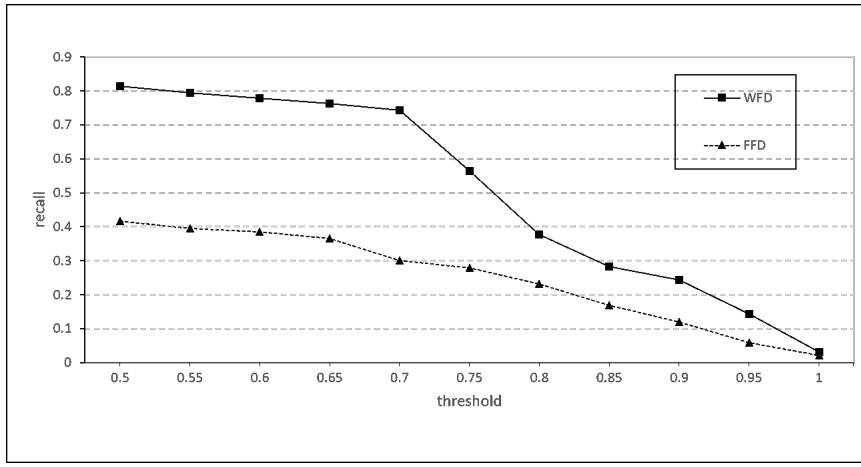


Figure 7. Recalls of FFD and WFD

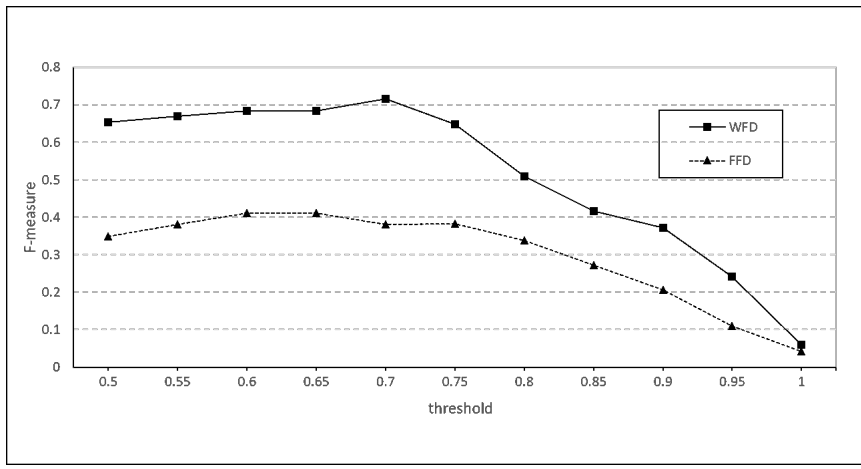


Figure 8. F-measures of FFD and WFD

that of FFD. Relatively, the FFD's recall decreases slowly with the increase of the threshold, while the WFD's recall has a sharp decline with threshold = 0.75. WFD relaxes requirements for uniqueness of key and strict inclusion dependency, therefore its recall is higher than FFD.

Generally, the precision and recall are mutually restricted, and F-measure is a combination of these two indicators which can inflect the overall performance of the method. From Figure 8, F-measure of WFD is higher than that of FFD under various threshold values. WFD and FFD with lower confidence thresholds (less than 0.7 and 0.65, respectively) could not achieve full effectiveness in detecting foreign key

candidates. Obviously, it is more possible to contain false positives in candidates with low thresholds. As the threshold grows, the F-measure of WFD and FFD reaches highest (0.72 for WFD and 0.41 for FFD). After that, the F-measure value decreases because recall decreases along with high confidence threshold.

In summary, our machine algorithm performs better than the state-of-the-art algorithm for discovering foreign key candidates on web tables.

6.2 Evaluation of Power of the Crowd

Analyzing the result made by WFD, we find that the false positive candidates are generated mainly because of lack of tables' semantics. With the help of the crowd, those false positives could be easily distinguished. The crowdsourcing experiment is implemented on CrowdSR [12], which is a crowdsourcing platform for semantic recovering of web tables.

From Figure 8, we can see that the F-measure of WFD reaches to the maximum value when the confidence threshold is set to 0.7. So, we set the threshold to 0.7 in the following experiment which means that 593 foreign key candidates including 409 true foreign keys and 184 false positive candidates need to be verified by the crowd. We create corresponding microtasks and post them to the crowd. These tasks are organized to 36 groups, each of which includes the true foreign keys and false positive candidates. Each task is finished by at least three workers with professional knowledge, and the majority voting are used to aggregate answers. Before doing tasks, workers should pass a qualification test which consists of three simple foreign keys verification tasks.

As a result, 376 candidates are verified as true foreign keys, among which 371 true foreign keys are correctly verified, and 5 false positive candidates are verified as true foreign keys by mistake. The performance comparison of machine algorithm and hybrid method is shown in Figure 9.

Obviously, the precision and F-measure of the human-machine hybrid method is better than the machine algorithm. The precision of the hybrid method is improved to $371/376 = 98.67\,\%$ from $68.97\,\%$ with the help of the crowd, since most candidates verified as true foreign keys by crowd are proved to be true. The recall of the human-machine hybrid approach is lower ($67.45\,\%$ vs. $76.36\,\%$) than that of the machine algorithm, because all foreign keys discovered by the human-machine hybrid method are contained in the output of machine algorithm.

6.3 Evaluation of Effectiveness of Dynamical Task Selection Method

This experiment is conducted on the same candidate set mentioned in Section 6.2, which contains 593 foreign key candidates with 409 true foreign keys and 184 false positive candidates. We use the combined task selection method to select the most valuable candidates to be verified and compare candidate numbers and precision with the naive task assignment method (i.e. assign all candidates to workers). Table 2 shows the result.

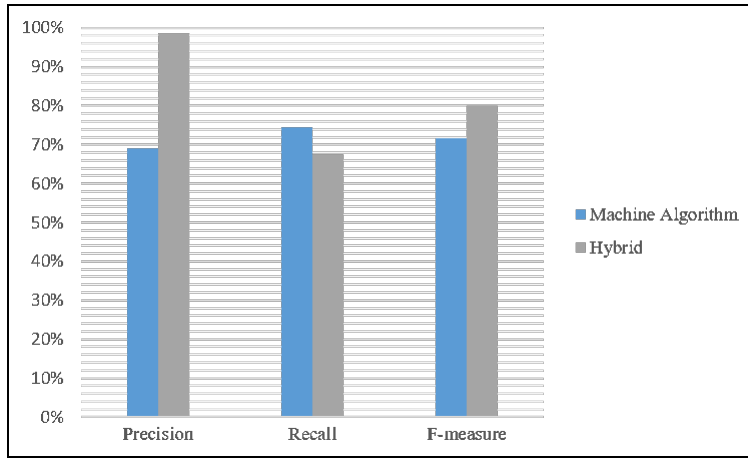


Figure 9. Performance comparison

Method	Crowdsourced Candidate Numbers	Foreign Keys Output	True Foreign Keys	Precision
Task Selection	322	384	368	95.83 %
Naive	593	376	371	98.67 %

Table 2. Naive task assignment method vs. task selection method

By the task selection method, total number of crowdsourced candidates is reduced to 322, about 54.30 % of the number of naive method. The task selection method outputs more foreign keys (384 vs. 376) because some candidates that are mislabeled as false positives in the naive method are deduced as true. There is no significant difference in the number of true foreign keys output by two methods (368 vs. 371). Conflict detection could help to remove candidates conflicted with the true foreign key, and inclusion dependency detection could help to deduce foreign key relationship. However, if candidates are mislabeled as true by the crowd, those candidates that are in conflict with them may be wrongly recognized as false positives. So, the precision of the task selection method is slightly lower (95.83 % vs. 98.67 %) than that of the naive method

To summarize these experimental results, our task selection method can effectively reduce the number of tasks thus reducing the monetary cost. Although some human errors may be amplified, foreign key detection precision under the task selection still reaches 95.83 %.

6.4 Evaluation of Effectiveness of the Task Reduction Method

Finally, we evaluate the task reduction method based on sampling strategy. Our main goal is to compare the time cost of the tasks reduced by our method with

the naive tasks without sampling. In order to evaluate the effectiveness of the task reduction method, we vary the sampling rate from 0.1 to 0.5, and compare the finish time and precision.

We sample the dependent table randomly, then partition the referenced table into two parts, the covered part and the uncovered part. From the covered part, we extract all tuples related to the sampled dependent table. Figure 10 shows the average time cost for each task with different sampling rate. Obviously, the average latency is substantially proportional to the sample rate. The time cost of our task reduction method under sampling rate from 0.1 to 0.5 is always lower than the average time cost 54 s of naive tasks.

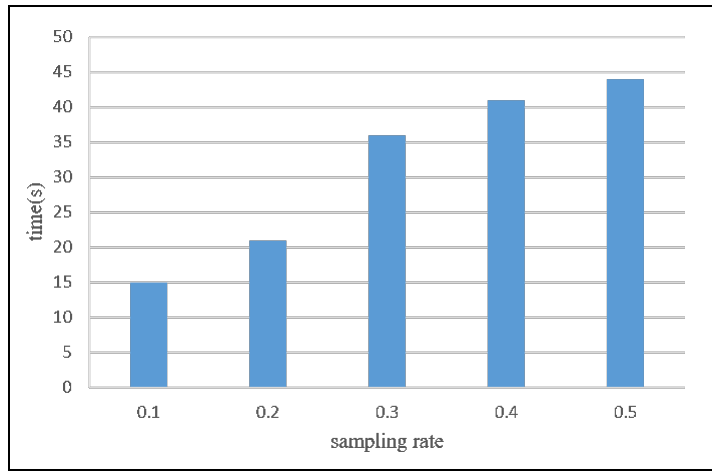


Figure 10. The average time cost under different sampling rate

Though our task reduction method keeps as much original table features as possible by using combinational sampling strategy, it is more easily for the crowd to make erroneous judgment when the table has been badly compressed with low sampling rate. Figure 11 shows the precision of the crowd verification result in different sampling rate. The precision increases with the increase of sampling rate. When the rate is set to 0.5, 364 candidates are verified as foreign keys by the crowd among which 353 are true foreign keys and the precision is close to the precision of the naive method. All the precisions of labeling results under sampling rate from 0.1 to 0.5 are more than 92 %.

In summary, our task reduction method with sampling strategy has a better performance than the naive method. Setting the sampling rate to 0.4, the average time cost is reduced to 41 s, while the precision of the verification result is very close to the precision of the naive method.

Quality vs. Cost. As both the precision and time cost are positively related to the sampling rate, there is a tradeoff between quality and cost. It is very im-

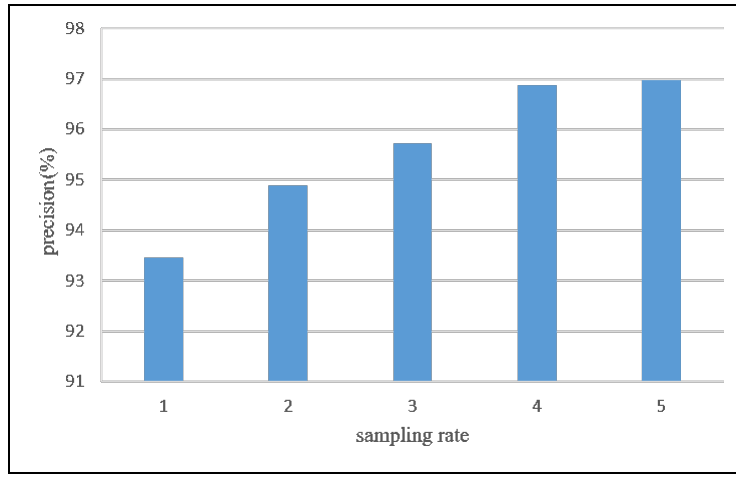


Figure 11. Precision comparison under different sampling rate

portant to select an appropriate sampling rate when using this task reduction method.

7 RELATED WORK

Recently, structured data from the web, such as the web tables, has been identified to have a high value in research. Thus, many researchers tend to recover [13, 14, 15] or integrate [16, 17] data in web tables.

Foreign key detection is an important work for analyzing and integrating data in web tables. Previous researches mainly focus on identifying inclusion dependencies. Bauckmann et al. proposed an algorithm named SPIDER for detecting unary inclusion dependencies [4], while some works [1, 2, 3] present a global way for foreign key detection in relational database. Rostin et al. detect putative foreign keys with a learning based method and propose some meaningful features for classifying inclusion dependencies [1], including DistinctDependentValues, Coverage, ColumeName, and DependentAndReferenced. Chen et al. propose a fast foreign-key detection method in PowerPivot [2] to perform this detection interactively and with high precision even when data sets scale to hundreds of millions of rows and the schema contains tens of tables and hundreds of columns. Randomness is proposed and an algorithm is developed to discover single-column and multi-column foreign keys in [3]. However, all the previous methods are not effective for discovering foreign keys on web tables of poor quality, which could not satisfy the entity integrity constraint and referential integrity constraint.

Fortunately, these problems could be solved easily with human's intelligence. Crowdsourcing is a good way to solve problems that are difficult for computers, and

it is widely used in academia such as entity resolution [7], sentiment analysis [8], and image recognition [9]. In this paper, we propose a hybrid human-machine framework to detect foreign keys on web tables. After discovering candidates and evaluating their confidence of being true foreign keys by machine algorithm, we verify those false positives leveraging the power of crowd.

8 CONCLUSIONS

Previous researches on foreign key detection mainly focus on finding the foreign key relationships between the relational tables in database. We are the first to propose a hybrid human-machine framework for discovering foreign keys on web tables which may not satisfy the entity integrity constraint and referential integrity constraint. To reduce the monetary cost, we proposed a dynamical task selection method based on conflict detection and inclusion dependency. Besides, to make workers complete tasks more effectively, sampling strategy is applied to reduce the task volume. The experimental results show our hybrid human-machine approach could indeed achieve a much higher detection precision with lower monetary cost and time cost.

Acknowledgment

This work is supported by the National Key R&D Program of China (2018YFC0809-800), and the National Natural Science Foundation of China (61370060, 61673048).

REFERENCES

[1] Rostin, A.—Albrecht, O.—Bauckmann, J.—Naumann, F.—Leser, U.: A Machine Learning Approach to Foreign Key Discovery. 12th International Workshop on the Web and Databases (WebDB 2009), Providence, Rhode Island, USA, June 2009.

[2] Chen, Z.—Narasayya, V.—Chaudhuri, S.: Fast Foreign-Key Detection in Microsoft SQL Server PowerPivot for Excel. Proceedings of the VLDB Endowment, Vol. 7, 2014, No. 13, pp. 1417–1428, doi: 10.14778/2733004.2733014.

[3] Zhang, M.—Hadjieleftheriou, M.—Ooi, B. C.—Procopiuc, C. M.—Srivastava, D.: On Multi-Column Foreign Key Discovery. Proceedings of the VLDB Endowment, Vol. 3, 2010, No. 1-2, pp. 805–814, doi: 10.14778/1920841.1920944.

[4] Bauckmann, J.—Leser, U.—Naumann, F.—Tietz, V.: Efficiently Detecting Inclusion Dependencies. 2007 IEEE 23rd International Conference on Data Engineering, 2007, pp. 1448–1450, doi: 10.1109/icde.2007.369032.

[5] Cafarella, M. J.—Halevy, A.—Wang, D. Z.—Wu, E.—Zhang, Y.: WebTables: Exploring the Power of Tables on the Web. Proceedings of the VLDB Endowment, Vol. 1, 2008, No. 1, pp. 538–549, doi: 10.14778/1453856.1453916.

[6] Google Fusion Table. Availaible at: `https://research.google.com/tables`.

[7] Wang, J.—Kraska, T.—Franklin, M. J.—Feng, J.: CrowdER: Crowdsourcing Entity Resolution. Proceedings of the VLDB Endowment, Vol. 5, 2012, No. 11, pp. 1483–1494, doi: 10.14778/2350229.2350263.

[8] Zheng, Y.—Wang, J.—Li, G.—Cheng, R.—Feng, J.: QASCA: A Quality-Aware Task Assignment System for Crowdsourcing Applications. Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD '15), 2015, pp. 1031–1046, doi: 10.1145/2723372.2749430.

[9] Welinder, P.—Perona, P.: Online Crowdsourcing: Rating Annotators and Obtaining Cost-Effective Labels. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Workshops (ACVHL), 2010, pp. 25–32, doi: 10.1109/cvprw.2010.5543189.

[10] Li, G.—Wang, J.—Zheng, Y.—Franklin, M. J.: Crowdsourced Data Management: A Survey. IEEE Transactions on Knowledge and Data Engineering, Vol. 28, 2016, No. 9, pp. 2296–2319, doi: 10.1109/TKDE.2016.2535242.

[11] Winkler, W. E.: String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. Proceedings of the Section on Survey Research Methods, 1990, pp. 354-359.

[12] Liu, H.—Wang, N.—Ren, X.: CrowdSR: A Crowd Enabled System for Semantic Recovering of Web Tables. In: Dong, X., Yu, X., Li, J., Sun, Y. (Eds.): Web-Age Information Management (WAIM 2015). Springer, Cham, Lecture Notes in Computer Science, Vol. 9098, 2015, pp. 581–583, doi: 10.1007/978-3-319-21042-1_67.

[13] Venetis, P.—Halevy, A.—Madhavan, J.—Paşca, M.—Shen, W.—Wu, F.—Miao, G.—Wu, C.: Recovering Semantics of Tables on the Web. Proceedings of the VLDB Endowment, Vol. 4, 2011, No. 9, pp. 528–538, doi: 10.14778/2002938.2002939.

[14] Wang, J.—Wang, H.—Wang, Z.—Zhu, K. Q.: Understanding Tables on the Web. In: Atzeni, P., Cheung, D., Ram, S. (Eds.): Conceptual Modeling (ER 2012). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 7532, 2012, pp. 141–155, doi: 10.1007/978-3-642-34002-4_11.

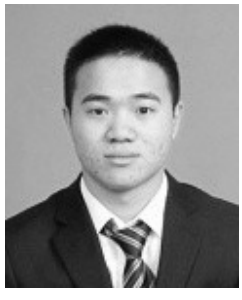
[15] Deng, D.—Jiang, Y.—Li, G.—Li, J.—Yu, C.: Scalable Column Concept Determination for Web Tables Using Large Knowledge Bases. Proceedings of the VLDB Endowment, Vol. 6, 2013, No. 13, pp. 1606–1617, doi: 10.14778/2536258.2536271.

[16] Yoshida, M.—Torisawa, K.—Tsujii, J.: A Method to Integrate Tables of the World Wide Web. Proceedings of the International Workshop on Web Document Analysis (WDA 2001), 2001, pp. 31–34.

[17] Gonzalez, H.—Halevy, A. Y.—Jensen, C. S.—Langen, A.—Madhavan, J.—Shapley, R.—Shen, W.—Goldberg-Kidon, J.: Google Fusion Tables: Web-Centered Data Management and Collaboration. Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD '10), 2010, pp. 1061–1066, doi: 10.1145/1807167.1807286.

Xiaoyu Wu is a Ph.D. candidate in computer science in Beijing Jiaotong University. Her main research areas include web data integration, big data management and data mining.

Ning Wang received her Ph.D. degree in computer science in 1998 from Southeast University in Nanjing, China. She is currently serving as Professor in School of Computer and Information Technology, Beijing Jiaotong University, China. Her research interests include web data integration, big data management, data quality and crowdsourcing.



Huaxi Liu received his Master degree in computer science from Beijing Jiaotong University in 2016 and currently works in Huawei Technologies Co., Ltd. His research interests include web data integration, data mining and crowdsourcing.