http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (1): 37-47, 2007
ISSN 1812-5638

FROut: A Novel Approach to Linking Large Databases

[1]Mingzhen Wei, [2]Andrew H. Sung and [3]Martha E. Cather
[1]P.O. Box 3209, Socorro, NM, 87801
[2]Department of Computer Science, 801 Leroy Place, Socorro, NM, 87801
[3]PRRC, 801 Leroy Place, Socorro, NM, 87801, USA

Abstract: The present study, deals with record linkage problems that occur when multiple data sources are different in size or are different in data formats or conventions, which is often seen in real practice. A systematic solution, *FROut* which stands for Filtering Relevant Out, is proposed. In the present study, a high quality filtering or searching strategy is the key to the success of record linkage practices. The importance of domain knowledge and data quality is emphasized for selecting the most reliable and important identifying attribute domains in the filtering strategy design. By generating different dynamic filtering criteria for the records processed, the new searching algorithm generates different sizes of relevant record sets, which ensures that all selected records are somehow relevant to the targeting records. By designing proper filtering criteria to consider only reliable data in identifying attribute domains, this approach saves a large number of wasteful comparisons in later stage of record linkage, hence improves the record linkage efficiency significantly. A linear relationship between computational cost and size of incoming data sets is observed, which is important for estimating the workload on different sizes of incoming data sets. The proposed approach was tested using real petroleum production data sets. Empirical results show that the new approach not only can resolve the problems that SNM-based methods cannot, but it also provides great computational efficiency and high linkage accuracy.

Key words: Data engineering, record linkage, data quality control, filtering relevant records, linkage decision model, filtering criteria, searching strategy

INTRODUCTION

Record linkage is defined as a process of comparing records from two or more data sources in order to determine which pairs of syntactically different records represent semantically same real-world entities. As knowledge discovery and data warehousing are becoming increasingly demanding, more and more relevant databases are involved in large-scale projects. Linking records in multiple sources becomes increasingly important. Since each data source is expected to contain various quality problems (Rahm and Do, 2000; Dasu and Johnson, 2003), developing record linkage is not a trivial task.

Record linkage can be applied in different sections to resolve a variety of complicated problems. For example, it is a fundamental task for businesses to collect correct and consistent costumer information in order to understand shopping behavior and it is a key operation in tracking a suspicious criminal to pull all records with varying identifying information together (Wang *et al*., 2004). Pioneer record linkage research was conducted in 1960s by Newcombe *et al.* (1959; 1962) trying to link individual facts to form an extensively documented history of a person's life. Newcombe studied problems and key components in linking individual personal information, like marriage, birth and hospital records. Detailed methods were used to process this personal information, but the theories of Newcombe *et al.* (1959; 1962) can be applied in other sectors.

In record linkage, the key components are: 1) a searching operation or algorithm to bring together relevant records in multiple data sources and 2) a comparison function to determine whether two candidate records are equivalent or not. Extensive research has been conducted on setting up efficient strategies to implement efficient searching and effective comparison. A searching algorithm is tightly associated with performance of linking both in accuracy (by bringing as many relevant records to compare) and efficiency (by constraining retrieval of as many irrelevant records). Hence it is important to select the appropriate searching strategy for specific record linkage problem.

Based on the concept of sorting and merging within a window, the Sort Merge Band Join proposed by Bitton and DeWitt (1983), Hernandez and Stolfo (1995; 1998)

Corresponding Author: Mingzhen Wei, P.O. Box 3209, Socorro, NM, 87801, USA Tel: 505-835-5171 Fax: 505-835-6031

set forth a series of methodologies to handle record linkage problems, named merge/purge in the studies. The approaches all start with an integrated data source from multiple sources. The Sorted Neighborhood Method (SNM) sorts the merged data source based on one selected identifying attribute, compares the neighboring records within a window of fixed size *w* and finishes the whole data source by sliding the window down gradually. The time complexity is O(N log N) if $w < \lceil \log N \rceil$, or O(wN) if $w \geq \lceil \log N \rceil$, where N is the number of records in the integrated source. Variations of SNM include Clustering SNM (C-SNM) and Multi-Pass SNM (MP-SNM). C-SNM clusters the integrated data source into several clusters and applies SNM in each cluster; MP-SNM executes independent SNM several times and uses a different sorting key in each run. These are very popular record linkage strategies used in data cleaning and data linkage (Lee *et al*., 1999; Monge and Elkan, 1997; Neiling and Muller, 2001; Winkler, 2001). But when two data sources greatly differ in size, or when two data sources have data convention differences in identifying attribute domains as in NAME<first name, last name> and NAME<last name, first initial>, it is difficulty or even impossible to bring together relevant records and to implement effective comparison.

In Information Retrieval (IR), each word or token is assigned a certain importance to represent the string or the document (Rahm and Do, 2000; Luhn, 1958). Word significance can be measured by the combination of word occurrence frequency and its occurring positions in sentences. Rank order of a word is the order of its occurring frequency. The relationship curve of frequency f and rank order r yields a hyperbolic curve as in Zipf's Law (Zipf, 1949). Two cutoffs were introduced to exclude non-significant words. The words exceeding the upper cutoff were considered too common and those below the lower cutoff too rare and therefore not contributing significantly to the content of the document (Fig. 5).

Inverse Document Frequency (IDF) weighting strategies reflect the word significance mentioned above and are widely applied in string or record string comparison. In string comparison, IDF weighting strategies have been effectively employed by Newcomb *et al.* (1959), Chaudhuri *et al*. (2003) and Ananthakrishna *et al.* (2002) to distinguish the importance of words in a document. Words with higher significance have higher influence in matching strings with various equivalence errors. However, the effective use of significant tokens in record linkage has not been considered earlier.

In record linkage, to design a robust searching algorithm to bring together relevant records, it is a must to obtain the most reliable and discriminative information from the identifying attribute domains. Identifying attribute domains serves to link records of same entity in different sources. Since data sources are expected to contain various data problems, errors are expected in all identifying attribute domains, even in very important ones, such as a person's social security number. Kukich (1992) reported that people rarely make same errors when they type, except for a few patterns. Therefore, words with an extremely low frequency (e.g., frequency = 1) probably, but not necessarily, indicate the existence of errors. From this point, the words with the smallest frequencies are not necessarily the most important words for representing the true values.

Coincidentally, based on Zipf's Law, the words with frequencies between the lower cutoff and upper cutoff are the most representative. These words are also the most reliable. After abstracting the most reliable and representative words from the identifying attribute domains, relevant records can be filtered out from a large data set to be compared with the targeting records. By doing this, the relevance of records selected is guaranteed by the representative words applied in the searching algorithm. By further enhancing the searching algorithm by combining all identifying attribute domains, using the reliable parts of their data, the efficiency of record linkage can be improved significantly by constraining the selection of irrelevant records.

Record linkage has also been studied using statistical methods, as by Jaro (1989), Elfeky *et al.* (2002), Fellegi and Sunter (1996), context information to improve the linkage accuracy as by Bhattacharya and Getoor (2004) or to clean spurious links as by Lee *et al.* (2004), employing a hierarchical graphical model framework under unsupervised machine learning as in Ravikumar and Cohen (2004), combining with Database Management Systems (DBMS) to compare the similarity of identifying information by Record Similarity (RS), proposed by Lee *et al.* (1999), or Q-grams as reported by Gravano *et al.* (2001), mapping records to a high-dimensional Euclidean space and finding similar object pairs within a certain threshold, proposed by Jin *et al.* (2003).

The selection of comparison functions is a domain-dependent task. Several comparison functions can be applied, such as edit distance, proposed by Levenshtein (1996) and its variations as in Winkler (1994), Jaro algorithm reported by Jaro (1989) and variations reported by Winkler (1994), a simple field matching algorithm and recursive version reported by Monge and Elkan (1996, 1997).

In the present study, a new record linkage strategy is developed based on the concept of significant words

and by taking advantage of the discriminative power of identifying attribute domains. The main contributions include: 1) a new record linkage strategy is proposed and studied; 2) a corresponding searching algorithm is developed; 3) performance of the new approach is studied from aspects of accuracy and efficiency; 4) the computation cost of new approach is studied and a linear model is revealed and 5) real petroleum production data sets are employed to test the proposed approach, with empirical results showing that the new approach outperforms the SNM-based methods in cases studied.

PROBLEM FORMULATION

A record linkage process contains three main components: 1) a searching algorithm to bring relevant records close, 2) a comparison function to compare resulting relevant records to determine their equivalency and 3) a linkage decision model to determine the linkage status of comparing pairs, as shown in Fig. 1.

Searching: A record linkage process starts from the manipulation of multiple data sources according to its searching algorithm. The searching algorithm needs to generate the potential linkable records in multi-sources which is highly demanded for improving the linkage accuracy and efficiency. To cut computational cost without losing linkage accuracy, a good searching algorithm is expected to generate as many relevant records as possible while constraining as many irrelevant records as possible from being selected. In the present study, a new searching algorithm was developed to fulfill the goal of linking data sources with high accuracy and efficiency.

String similarity: Similarity functions are applied to evaluate the equivalence of attribute values and records. Many similarity functions can be applied; their selection is a domain-dependent task. In the present study, edit distance, a very commonly-used string similarity metrics is applied. Edit distance, also termed as Levenshtein distance, measures the minimum number of edit operations (insertions, deletions and substitutions) that are required to transform one string to the other. For example, ED ("university", "univercity") = 1. By its nature, edit distance fits for comparing strings collected by OCR scanning processes.

Record similarity: Given two relations R and S that share a set of attributes A_1, …., A_p, record strings can be constructed by concatenating the string value of each selected attribute domain for a specific record, as in the following formula:

$$\text{str Record} = \text{concatenate}(\text{strA}_1, \text{strA}_2, \ldots, \text{strA}_p) \quad (1)$$

where strA_1, strA_2, …, strA_p are values of attribute domains of A_1, A_2, …, A_p, respectively for a specific record. Then the strings from relations R and S are compared using edit distance metrics:

$$\text{record Similarity} = \text{ED}(\text{R.str Record}, \text{S.str Record}) \quad (2)$$

Fuzzy linkage rules: For making the final linkage decision, results from record similarity comparison are considered. Fuzzy decision rules are applied to classify record pairs into three disjoint sets:

- Linked set: record pairs that are linked candidates, filtered by ED (R.str Record, S.str Record) ≤ Lower;
- Unlinked set: record pairs that cannot possibly be linked, filtered by ED (R.str Record, S.str Record). > Upper;
- Undecided set: record pairs that are uncertain as to their linking status filtered by Lower < ED (R.str Record, S.str Record) ≤ Upper.

Where Lower and Upper are edit-distance thresholds for linked records and unlinked records, Lower < Upper and they are assigned based on the domain knowledge.

Performance evaluation: As in information retrieval processes, performance of record linkage is evaluated by linkage accuracy (recall and precision) and computation cost, taking record linkage as retrieving relevant or equivalent records from multiple sources. All information retrieval results can be classified into the following categories: a) retrieved relevant information represented

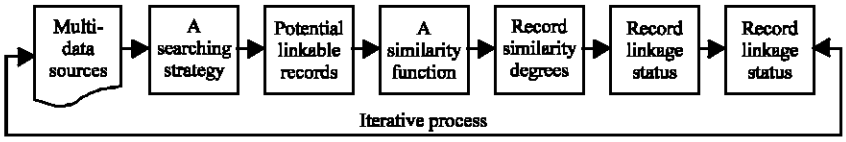

Fig. 1: Diagram of record linkage process

by A; b) retrieved irrelevant information, represented by B; c) unretrieved relevant information, represented by C and d) unretrieved irrelevant information, represented by D. Recall and precision are defined as follows:

$$\text{recall}: R = \frac{A}{A + C} \tag{3}$$

$$\text{precision}: P = \frac{A}{A + B} \tag{4}$$

In the present study, linkage results are assigned in three disjoint sets: linked L, unlinked UL and undecided UD. Total retrieved relevant records are in both linked set L and undecided set UD. Use A_P as the number of retrieved relevant records in set P. Following formula can be used to calculate the total retrieved relevant records in a linkage process:

$$A = A_L + A_{UD} \tag{5}$$

Computation cost is evaluated using the total cost and most expensive components. It is also compared with previous record linkage processes, like SNM-based processes.

RECORD LINKAGE PROBLEM ENCOUNTERED

As discussed earlier, the majority of previous record linkage practices integrates multiple data sources into one and then sort this single source based on selected identifying attribute domain(s). These studies assume that: 1) by proper pre-process, as suggested by Hernandez and Stolfo (1995), Lee *et al.* (1999) and Monge and Elkan (1997), multiple sources can be successfully merged into one; and 2) the sorting process can bring syntactically different representations of same real-world entities into a small-size neighborhood. These assumptions are implied in all SNM-based approaches.

Two disadvantages, however, are associated with these assumptions. First, it is difficult or, at least, extremely time-consuming to achieve these pre-process goals for various reasons, such as typographical errors, misspellings, phonetic problems, character recognition problems, different data conventions and data formats caused by data flow, proposed by Dasu and Johnson (2003). With all these unclean data problems, bringing equivalent values close enough into a small size window to compare becomes complicated. Thus, there are inherent problems in those methods.

Besides the equivalence errors, multiple data sources are expected to be different in size. If one source is larger (50 times or more) than the others in size, the resulting data file after merging these data sources is expected to be dominated by one source. After sorting, the majority of records to be compared that fall into a fixed window size will be biased by the large source. Record linkage will encounter a large number of wasteful comparisons. If a super-large window size is applied, such as 1000 or larger, the computational cost will be increased significantly.

This case is relatively common in real-life systems. For example, the daily data management of a company includes cleaning incoming data sets based on its knowledge base data sets, the reference sources in record linkage or the outer sources in data cleaning. The outer sources become larger as they collect standardized data with the progress of businesses. In these cases, it is not suitable to merge all data sources together. This paper explains a new strategy that we developed to handle these situations.

NEW RECORD LINKAGE STRATEGY

The novelty of our solution lies in the proposed searching algorithm and corresponding strategy of multiple data sources.

Proposed searching algorithm: Figure 2 shows the diagram of a proposed searching algorithm, the kernel of a new record linkage solution. It can be seen that the incoming data source is not integrated with the reference data source, as in SNM-based methods. For each record in the incoming data source, a filtering criterion is constructed to search relevant and potential linkable records in the reference data source (Fig. 2). Only records in the reference source that satisfy the filtering criteria are selected to be compared.

To ensure that a proper filtering criterion can be derived from each record in the incoming data source, extensive work is required to develop a proper filtering strategy. To fulfill the goals of searching algorithms, the filtering criteria are expected to be both reliable and discriminative. Therefore understanding each identifying attribute domain is a must.

Domain knowledge and data profiling: In order to select proper identifying attributes in designing filtering criteria, it is necessary to investigate the domain knowledge associated with the incoming data source for two purposes: 1) to select the most important and most discriminative attributes possible and 2) to understand the possible data conventions in each attribute domain

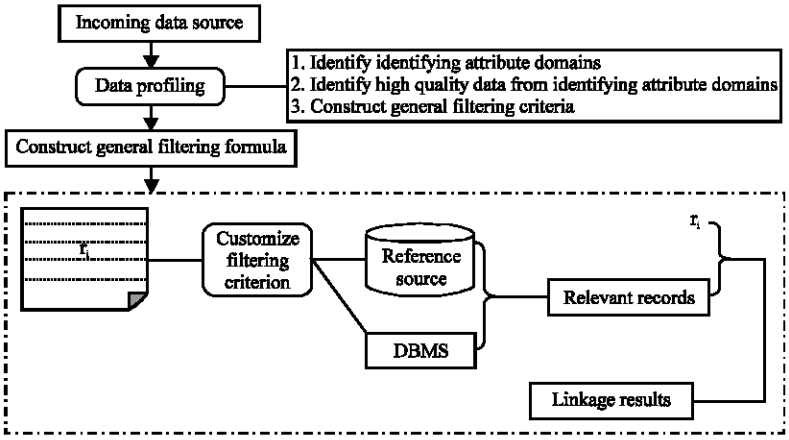

Fig. 2: Diagram of proposed searching strategy

and the data quality rooted in the nature of each domain. For example, a relation containing student records would consist of the following attributes: [student name], [student ID], [SSN], [registration ID], [semester ID], [sex], [contact] and etc. Among these attributes, [student name], [student ID] and [SSN] are the most important identifying attributes for students. It is generally believed that numeric-string data, such as phone number or SSN, are less problematic than string-valued data. In this example, [student ID] and [SSN] could be the most reliable attributes for linking records in other databases. Now we consider another example. Customer information could be represented by [first name], [last name], [address 1], [address 2], [city], [state], [phone number] and etc. It is obvious that these properties are all important for identifying the same customers. Domains [first name] and [last name] are expected to contain the most important identifying information but others are slightly less significant because they are subject to change over time. Among these attributes, [last name], [city], [state] and [phone number] are expected to be more accurate and reliable. In order to know the exact data quality picture of each attribute domain, it is a must to apply corresponding techniques to analyze the data.

Data profiling is the systematic summarization of the content of a data source. Data profiling focuses on the instance analysis of individual attribute domains. It obtains general statistical information such as the data type, length, value range, discrete values and frequency, variance, uniqueness, occurrence of null values and typical patterns and provides an exact picture of data quality of a specific attribute domain from various aspects. Rahm and Do (2000) provide an excellent summarization on profiling for analyzing data quality from different aspects. Selecting proper profiling is also a domain-dependent work. Sometimes it is necessary to profile the sub-domains; data abstracted from individual domains to understand the data quality in more depth.

Filtering strategy: A filtering criterion is derived from the filtering strategy for each record in the incoming data source in the searching process. According to the goals of a searching algorithm, filtering criteria must be reliable and sufficient to bring as many relevant records as possible and to be selective to constrain as many irrelevant records as possible from being selected. In filtering strategy design, it is important to consider all identifying attributes and their discriminative power and data quality, based on the domain knowledge known and the results from data profiling. An efficient filtering strategy can be developed by selecting the most discriminative identifying attributes and abstracting reliable information from these attributes.

Given selected identifying attributes A_1, A_2, .., A_i and functions f_1, f_2, …, f_i to abstract reliable parts from those attributes, the filtering criteria are designed as follow:

$$\begin{aligned} \text{filter} = {} & f_1(A_1) && \text{if } f_1(A_1) \neq \text{null} \\ & \wedge f_2(A_2) && \text{if } f_2(A_2) \neq \text{null} \\ & \wedge f_3(A_3) && \text{if } f_3(A_3) \neq \text{null} \\ & \cdots\cdots && \\ & \wedge f_i(A_i) && \text{if } f_i(A_i) \neq \text{null} \end{aligned} \tag{6}$$

The formula can be interpreted as follows: for any given record, all important identifying properties should be considered as part of a filtering criterion when the reliable data is available and all these discriminative data should be *AND*ed logically to generate the most discriminative record set in record retrieval. It can be interpreted in SQL-like representation as:

SELECT attributes FROM reference_ table WHERE $f_1(A_1)$!= NULL AND A_1 contains $f_1(A_1)$ AND $f_2(A_2)$!= NULL AND A_2 contains $f_2(A_2)$ AND ... AND $f_j(A_j)$!=. NULL AND A_j contains $f_j(A_j)$

Fig. 3: SQL-like filtering criteria

where the contains operation means that $f_i(A_j)$ can be found in attribute A_i (Fig. 3). Functions f_1, f_2, …, f_i can be defined many ways. They can be data transformation formula, online user defined functions and offline functions or subroutines, whatever can abstract reliable information from selected attributes.

Overall record linkage algorithm: Following the basic record linkage procedure in Fig. 1, the overall record linkage algorithm is presented more clearly and explicitly in Fig. 4.

As discussed earlier, designing filtering strategy for specific application, is the key to the success of record linkage. It is intensively domain-dependent in order to select the proper identifying attribute domains and to design the proper functions for abstracting the most reliable and discriminative information from those domains in a filtering strategy. With comprehensive understanding of each domain and overall definition of databases and of the purposes of applications, high performance can be achieved.

Input: Incoming data source I with N records
Reference data source R with M records, usually M>>N
Edit distance as similarity metrics to compare string values
Output: Linked candidate set L
Unlinked set UL
Undecided set UD

Procedure:

1. Investigate the domain and its data quality for every identifying attribute domain;
2. Design filtering strategy, including selecting reliable identifying attribute domains and designing functions to abstract reliable and discriminative information from them;
3. Access records in I to be processed;
4. Link records in I and R:
   For each record r in I:
   1) Generate a filtering criterion based on designed filtering strategy;
   2) Search relevant records in R using the filtering criterion and generate a potential linkable record set S;
   3) Compare r and each record rs in S and obtain the edit distance ED:
   If ED≤Lower, put r and rs into L;
   If ED>Upper, put r and rs into UL;
   If Lower<ED≤Upper, put r and rs into UD.
5. further process

Fig. 4: The new record linkage algorithm

RESULTS AND DISCUSSION

Data sets and setups: Real data sets were employed to study the performance of the new searching strategy. The data sets are oil/gas production data collected by OCR scanning. The production records are coded as <[property name], [well ID], [township], [range], [section], [unit letter], [prod year], [pool ID], ……>. Among these properties, [property name], [well ID], [township], [range], [section] and [unit letter] are designed to represent the entity of oil/gas producers; [prod year] and [pool ID] and other production data are to record the oil/gas production data. In the database, the primary key would be [API], [pool ID] and [prod year]. The record [API] is associated with the entity of producers. One serious problem is that all previous production records in Annual Reports of the New Mexico Oil and Gas Engineering Committee had no APIs, because APIs were not available at that time. To integrate all production data into the current database, it is critical to identify correct APIs for all previous producers. OCR scanning processes were set up to collect data from the Annual Reports. There were two problems in finding correct APIs : 1) OCR scanning processes were expected to encounter some character recognition problems, which would cause difficulties in value matching and 2) some property values were expected to change over the production history. For example, the property name of a production well might be changed when its owner changed.

To assign correct APIs for producers in New Mexico, the Oil and Natural Gas Administration and Revenue Database (ONGARD) was employed as the reference source. ONGARD has more than 90,000 records and it is growing as new production wells are drilled. It is a relatively official and standardized database.

Experiments in this section were conducted for following purposes:

- To evaluate the dependence of data quality of selected identifying attributes and final linkage performance by trying different combinations of identifying attributes in filter design;
- To study the effectiveness of the new searching strategy compared with a SNM-based approach;
- To examine the effect of similarity thresholds on the linkage accuracy in the decision model;
- To determine the relationship of computation cost and the size of the incoming data source.

Data profiling and filtering criteria design: As mentioned earlier, [property name], [well ID], [township], [range], [section] and [unit letter] are all components

Table 1: Domain attributes and corresponding data characteristics

Attribute	Data length	Data representation	Data problems
Property name	≈ 30	Multiple words, numbers, characters	Character recognition errors, word transposing, changing value
Well ID	≤ 5	$[0\text{-}9]^+$, $[A\text{-}Z][0\text{-}9]^+$, $[0\text{-}9]^+[A\text{-}Z]$	Inconsistent representation, character recognition errors, missing values
Township	≤ 5	$[0\text{-}9]\{1,2\}[NS]^+$	Character recognition errors, missing values
Range	≤ 5	$[0\text{-}9]\{1,2\}[NS]^+$	Character recognition errors, missing values
Section	≤ 5	$[0\text{-}9.]^+$	Character recognition errors, missing values
Unit letter	≤ 3	$[0\text{-}9A\text{-}Z]^+$	Character recognition errors, missing values

Table 2: Basic profiling characteristics of [township], [township dir], [range], [range dir] and [section] for sample data set (Highlighted columns show the correct information for specific attribute domain and others show the basic profiling characteristics)

Attribute domain	Data type	# of distinct values	Invalid type	# of distinct invalid values	Correct values
Township num	Numeric	19	Null, character, string	6	0-33
Township dir	Character	6	Null, error, string	5	[N,S,n,s]
Range num	Numeric	32	Null, character, string	18	0-39
Range dir	Character	14	Numeric, null, string	13	[E, W,e,w]
section	Numeric	47	Null, string, character	10	0-36

Table 3: Basic profiling characteristics of [Township Num], [Range Num] and [Section] for sampled data set

Attribute	# of non-numeric values	# of null value	# of distinct non-numeric values
Township num	13 (0.5%)	2 (0.1%)	6 (31%)
Range num	52 (2.1%)	11 (0.5%)	18 (56%)
Section	19 (0.8%)	9 (0.4%)	10 (21%)

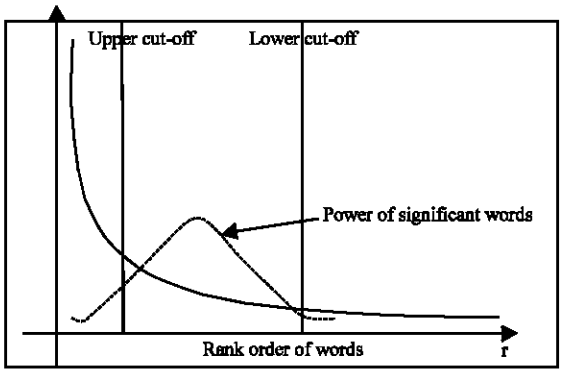

Fig. 5: Relationship between frequency of words and their rank order stated in the Zipf's Law

of identifying oil/gas producers. To understand their data qualities more clearly, the main characteristics are summarized as in Table 1.

When selecting proper identifying attributes to construct a proper filtering strategy, it is important to consider these characteristics of domains. Since numeric data and string data are mixed together in [well ID] and [unit letter] and are hard to separate, they are not selected in designing filtering criteria.

In order to abstract reliable data from selected identifying attributes, different strategies were developed. To abstract reliable and discriminative information from [property name] domain, Zipf's Law and concept of significant word are employed. As shown in Fig. 5, words with extremely large or extremely small frequencies cannot represent the document; while words with frequencies between upper cut-off and lower cut-off contain the most representative meanings of a document or a value.

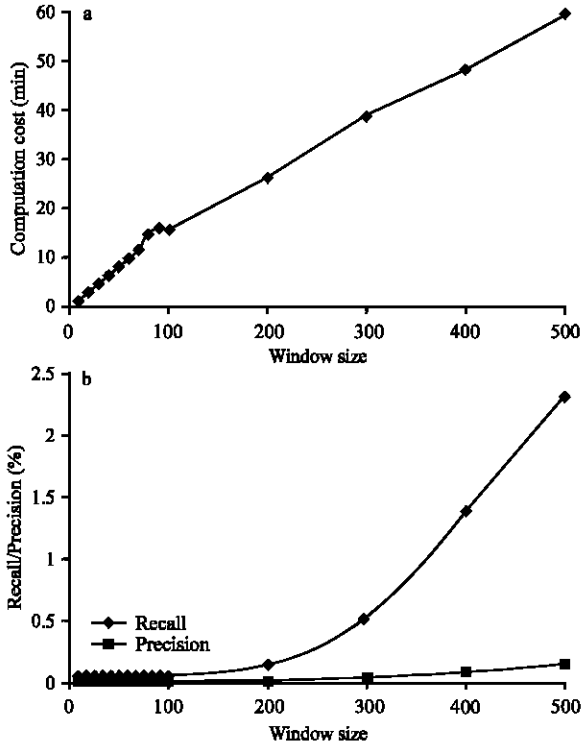

Fig. 6: Relationships between window size and recall, precision and computational cost using the SNM-based record linkage process

Since numeric data is less problematic than character data, the numeric parts of [township], [range] and [section] were considered for application in filtering strategy design. Data transformations were applied to separate these domains into numeric and string-valued parts. Further profiling, Table 2, shows that numeric data are more discriminative and more reliable than string-value parts. Their numeric parts, [township num], [range num] and [section] respectively, were profiled to gain deeper insight into their data quality, (Table 3).

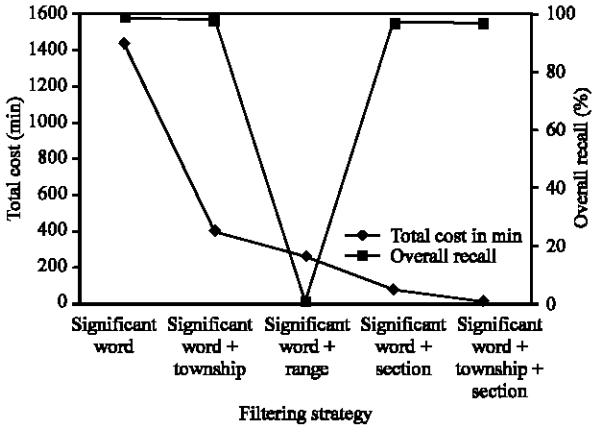

Fig. 7: Total computation cost and overall recall under different filtering strategies. Results are obtained under Lower = 5 and Upper = 8

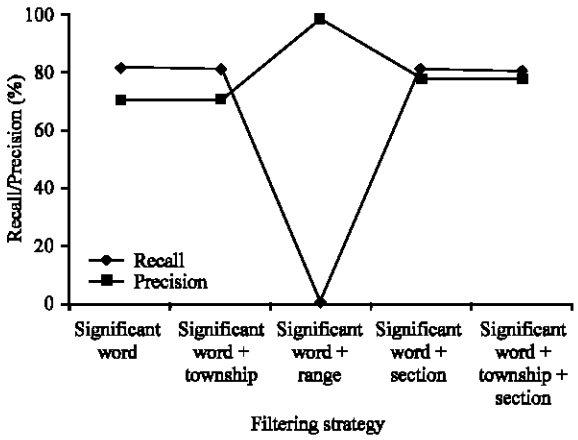

Fig. 8: Recall and precision of L under different filtering strategies. Results are obtained under Lower = 5 and Upper = 8

From the profiling results, [range num] is the most problematic. Experiment results using [range num] show very low linkage accuracy (Fig. 7). Based on the above investigation results, the filtering criteria were constructed with the following formula:

$$\begin{aligned}\text{filter}([\text{priducer's entity}]) = \text{significant}([\text{property name}]) \\ \wedge\ \text{numeric}([\text{township num}]) \\ \text{when numeric}([\text{township num}]) \neq \text{null} \\ \wedge\ \text{numeric}([\text{section}]) \\ \text{when numeric}([\text{section}]) \neq \text{null} \quad (7)\end{aligned}$$

The overall performance of the new approach is compared with a SNM-based strategy. The SNM-based experiments were implemented as follows: 1) merging incoming data source and reference source into one; 2) sorting the resulting data source based on [township num], [section] and [property name] and 3) selecting a fixed size of window to run SNM, using edit distance as the similarity metrics. Figure 6 shows that although the SNM-based approach is fast, especially when the window sizes are smaller than 100, the recalls are unexpectedly low. The recall cannot go up to 3% even when the window size is up to 500, in which case the computational cost is pretty high. These results confirm that SNM-based approaches are not suitable for record linkage in the cases presented.

Effect of filtering strategy: Figure 7 shows the significant effect of a filtering strategy, both on linkage accuracy and computation cost. Data in [range num] is not applied in the final filtering strategy, but is employed to show the effect of data quality of filtering strategies on the linkage accuracy.

With different filtering strategies, the linkage process can be extremely low (up to 24 h), but it can also be quite fast taking only about 20 min. This is because a filtering strategy determines whether a selective and reliable filtering criterion can be derived for a given record. If a filtering strategy is not sufficiently discriminative, such as only applying significant words from the [property name] domain, a large number of records will be selected for each given record, averaging 96053 records for one record in this case. This leads to large workload for relevant record access and record comparison. As the filtering strategy is enhanced by other discriminative identifying information, like [township num], [range num] and [section], computation cost decreases significantly. Computation costs for different filtering strategies also reveal that [section] is the most discriminative information used in the filtering strategy. This is because it is originally numeric domain so no transformation is required. This confirms that numeric data have higher quality than string-valued data.

Results of recall and precision show that when a filtering strategy is enhanced by reliable discriminative data, the linkage accuracy will not decrease or only decrease only slightly, with a large decrease in computational cost. As in Fig. 7, when the filtering strategy is enhanced by [township num] and [section], the overall linkage accuracy only changes from 99.2 to 98.3 and 97.7%. But when the filtering strategy is enhanced by [range num], which has the worst quality of all, the overall linkage accuracy changes significantly from 99.2 to 2%. This result shows the significance of reliable domains and reliable data applied in a filtering strategy. It also reflects the importance of pre-work for understanding the attribute domains and their data.

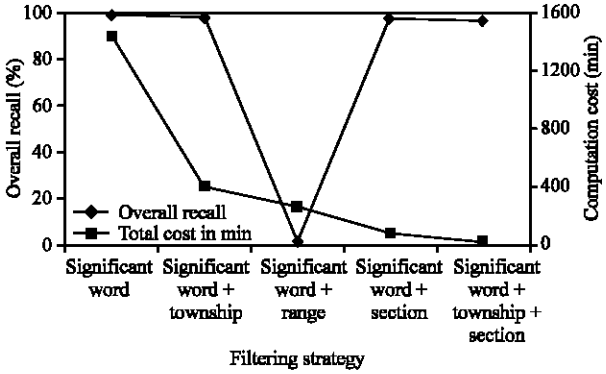

Fig. 9: Recall and precision of UD under different filtering strategies. Results are obtained under Lower = 5 and Upper = 8

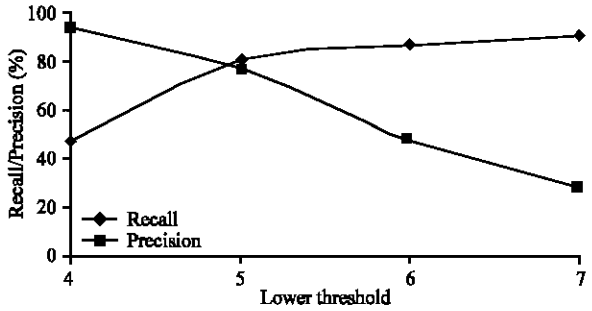

Fig. 10: Relationships between lower similarity threshold versus recall and precision (Upper = 8) in the linked set L.

Figure 8 shows the recall and precision in the linked set L under different filtering strategies. With reliable attribute domains and reliable data applied in the filtering strategies, recalls are in the same level, about 80%. But if some unreliable data is applied in the filtering strategy, as in the third strategy, recall goes down almost to zero. It indicates that unreliable data cannot be applied in our approach to efficiently filter out relevant records and greatly decreases the linkage performance.

Figure 9 shows recall and precision in the undecided set UD under different filtering strategies. Recall and precision in UD follow the tendency with respect to the filtering strategies. Strategies developed with reliable attribute domains and reliable data achieve higher recall; otherwise the recall is very low.

Effect of similarity thresholds: As in the record linkage decision model, Lower and Upper play important roles in determining the status of linking record pairs. Hence it is important to define proper thresholds for specific record linkage practice. From the decision model decreasing the

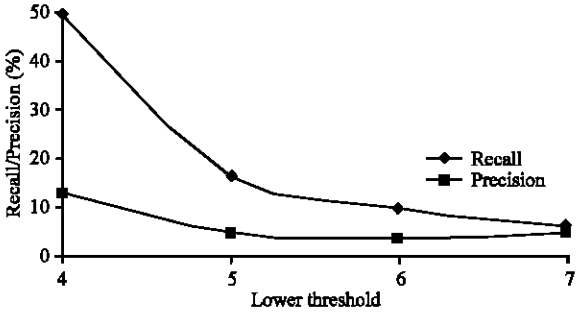

Fig. 11: Relationships between lower similarity threshold versus recall and precision (Upper = 8) in the undecided set UD

Lower similarity threshold will decrease the recall in a linked set; decreasing the difference between Lower and Upper thresholds will decrease the recall in the undecided set and decreasing the Upper threshold will increase the relevant records assigned in the unlinked set, which is not expected in the record linkage process. Therefore, it is a must to investigate the domain and data quality before these similarity thresholds can be properly assigned.

With the fixed Upper similarity threshold as eight, an overall recall of 97% is achieved. An overall precision is 0.87%. The lower precision indicates that a large number of wasteful records are selected by the searching algorithm; while the high recall proves that the current strategy is efficient in searching all relevant records. Figure 10 and 11 show the results of recall and precision in the linked set and undecided set.

Since the linked set L is the main set for analyzing the record linkage accuracy, it is important to improve the recall of relevant records in L. From the decision model, Lower threshold is the main parameter that affects linked set L. From Fig. 10, L can obtain high recall and relatively high precision by selecting a proper Lower threshold. A lower threshold puts tighter constraints to identify equivalent records and a higher threshold looses the constraints and lets more non-equivalent records get in, which leads to the decreasing of precision. Since post-processing is always expensive for information retrieval practice, finding proper tradeoff of recall and precision is important. In this case, the best tradeoff of recall and precision was achieved when Lower = 5, in which more than 80% of equivalent records are retrieved and about 80% of retrieved records are equivalent.

Some equivalent record pairs can be identified from the undecided set UD. Both Lower and Upper affect the recall and precision in the UD. Decreasing the Lower will make the recall higher, with more equivalent record pairs being assigned as undecided.

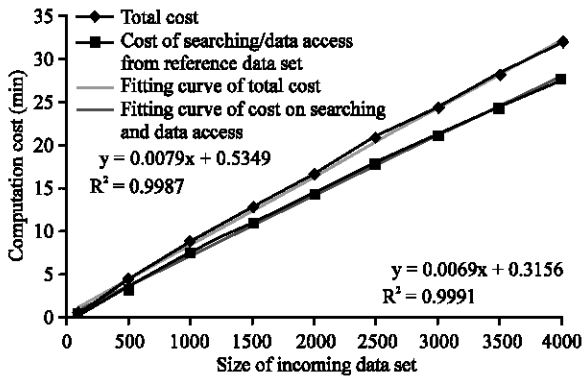

Fig. 12: Relationships between total computational costs of searching and data access from reference data set and the size of incoming data sets. Results obtained when Lower = 5 and Upper = 8

Effect of size of incoming data sets: Since the new approach implements the record linkage by filtering relevant records from other sources based on the given record in incoming data sources, the size of incoming data sources will affect the computation cost significantly. The larger the incoming data set, the higher the total computation cost. For each individual linkage, the computation cost is mainly for searching relevant records in the large reference data set $T_{Searching}$, access those records from the database server T_{Access} and string comparison $T_{recordCompare}$ and other process T_{other}. If the size of the incoming data set is *N*, total workload is

$$T(N) = \sum_{i=1}^{N} [T_{Searching} + T_{Access} + T_{RecordCompare} + T_{Other}]_i$$

It is a function of N.

Taking incoming data sets of sizes 100, 500, 1000, 1500, 2000, 2500, 3000, 3500 and 4000, computation cost results, either total cost or cost on the data searching and access, are shown in Fig. 12. From the fitting equations, cost on searching relevant records and access of those records dominates the overall computation cost, even up to 88% in some cases. Total cost follows equation T (N) = 0.0079N + 0.5349, where *N* is the size of incoming data set. This indicates, under the reference data set and the domain we are working on, that every increase of 1000 records in the incoming data set leads to an increase in computational cost of eight minutes. The cost for searching and accessing those records from the reference data set follows equation T (N) = 0.0069N + 0.1356, which indicates that a 1000-record increase in the incoming data set leads to a seven-minute increase in the cost of searching and accessing relevant records from the reference data set.

By this cost model, one can roughly estimate the cost when different sizes of incoming set are processed.

CONCLUSIONS

Record linkage is a powerful tool both for cleaning dirty data sources and linking relevant data sources to come up with a more comprehensive profile of an entity. In the present study, a new approach, FROut, is proposed to link multiple data sources efficiently and accurately, especially when involved data sources have large difference in sizes or when multiple data sources are different in formats or data conventions. The novelty is that only relevant records are selected based on a filtering criterion for given records derived from a proper filtering strategy. This is the strong point of the new approach and cannot be achieved by other methods. This approach improves the record linkage performance by constraining the access of irrelevant records and improving the linkage accuracy. With an enhanced filtering strategy that considers all reliable information in the most important identifying attribute domains, the filtering criterion for a given record, used in a large data set, can be very selective. The computation cost is improved. Through experiments with the real data sets, the following conclusions are drawn:

- As in other data cleaning process, a comprehensive understanding of each attribute domain in the incoming data set and an overall picture of the data set are very important;
- In designing filtering strategies, there are two key steps to make record linkage successful: 1) selecting the most important and discriminative identifying attribute domains and 2) designing proper functions for these domains that will abstract the most reliable information from them. By observing these steps, the probability of accessing *all* relevant records is increased and retrieval of irrelevant records decreases significantly. Performance can be enhanced greatly with high accuracy and high computational efficiency.
- With showing high recall and precision in linking real large data sets with low computational cost, the new approach was proven to be effective and efficiency;
- The cost of the new approach is dominated by relevant record access from the reference data source and the cost of record comparison. Empirical results show linear relationships between the total cost, cost of searching and data access from the reference

source and the size of incoming data sets. Both costs heavily depend on the average size of potential linkable record sets generated by searching processes, which is principally decided by domains and databases. With the model, it is easy to estimate the overhead of linkage practice, based on known experience.

REFERENCES

Ananthakrishna, R., S. Chaudhuri and V. Ganti, 2002. Eliminating fuzzy duplicates in data warehouses. In the Proceeding of the 28th VLDB Conference, Hong Kong, China.

Bhattacharya, I. and L. Getoor, 2004. Iterative record linkage for cleaning and integration. In Proceeding of DMKD'04, Paris.

Bitton, D. and D.J. Dewitt, 1983. Duplicate record elimination in large data files. ACM Transactions on Database Systems, Vol. 8: 255-265.

Chaudhuri, S., K. Ganjam, V. Ganti and R. Motwani, 2003. Robust and efficient fuzzy match for online data cleaning. In the Proceeding of the 2003 ACM SIGMOD Intl. Conf. on Manage. Data, San Diego, California.

Dasu, T. and T. Johnson, 2003. Exploratory Data Mining and Data Cleaning. Wiley-Interscience.

Elfeky, M.G., V.S. Verykios and A.K. Elmagarmid, 2002. TAILOR: A record linkage toolbox. In the Proceeding of the 18th ICDE, San Jose, CA.

Fellegi, I.P. and A.B. Sunter, 1996. A theory for record linkage. J. Math. Stat. Asso., 64: 1183-1210.

Gravano, L., P.G. Ipeirotis, H.V. Jagadish, N. Koudas, S. Muthukrishnan and D. Srivastava, 2001. Approximate string joins in a database (almost) for free. In Proceeding of the 27th VLDB Conference, Rome, Italy.

Hernandez, M.A. and S.J. Stolfo, 1995. The merge/purge problem for large databases. In Proc. of ACM SIGMOD/PODS, San Jose, CA.

Hernandez, M.A. and S.J. Stolfo, 1998. Real world data is dirty: Data cleansing and the merge/purge problem. J. Data Mining and Knowledge Discovery, Vol. 2.

Jaro, M.A., 1989. Advances in record-linkage methodology as applied to matching the 1985 Census of Tampa, Florida. J. Am. Stat. Assoc., Vol. 84.

Jin, L., C. Li and S. Mehrotra, 2003. Efficient record linkage in large data sets. In Proceeding of the 8th International Conference on Database Systems for Advanced Applications (DASFAA).

Kukich, K., 1992. Techniques for automatically correcting words in text. ACM Computing Surveys, Vol. 24.

Lee, M.L., W. Hsu and V. Kothari, 2004. Cleaning the spurious links in data. IEEE Intelligent Syst., 19: 28-33.

Lee, M.L., H. Lu, T.W. Ling and Y.T. Ko, 1999. Cleansing data for mining and warehousing. In the Proceeding of the 10th DEXA.

Levenshtein, V., 1966. Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady, Vol. 10.

Luhn, H.P., 1958. The automatic creation of literature abstracts. IBM J. Res. Develop., Vol. 2.

Monge, A.E. and C.P. Elkan, 1996. The field matching problem: algorithms and applications. In the Proceeding of the 2nd SIGKDD, Portland, Oregon, USA.

Monge, A.E. and C.P. Elkan, 1997. An efficient domain-independent algorithm for detecting approximately duplicate database records. In Proceeding of the ANM-SIGMOD Workshop on Research Issue on Knowledge Discovery and Data Mining, Tucson, AZ.

Neiling, M. and R.M. Muller, 2001. The good into the pot, the bad into the crop. preselection of record pairs for database fusion. In Proceeding of the First International Workshop on Database, Documents and Information Fusion, Magdeburg, Germany.

Newcombe, H.B., J.M. Kennedy and C. River, 1962. Record linkage: Making maximum use of the discriminating power of identifying information. Communications of the ACM, Vol. 5.

Newcombe, H.B., J.M. Kennedy, S.J. Axford and A.P. James, 1959. Automatic linkage of vital records. Science, Vol. 130.

Rahm, E. and H. Do, 2000. Data cleaning: Problems and current approaches. IEEE Data Eng. Bull.

Ravikumar, P. and W.W. Cohen, 2004. A hierachical graphical model for record linkage. In the Proceeding of the 20th conference on Uncertainty in Artificial Intelligence. Banff, Canada.

Wang, G., H. Chen and Atabashsh, 2004. Automatically detecting deceptive criminal identities, Communications of ACM, Vol. 47.

Winkler, W.E., 1994. Advanced methods for record linkage. Proceeding of the Section on Survey Res. Meth. Am. Stat. Assoc., pp: 467-472.

Winkler, W.E., 2001. Quality of very large databases, Technical Report RR/2001/04, Stat. Res. Report Series. US Bureau of the Census, Washington DC.

Zipf, H.P., 1949. Human Behavior and the Principle of Least Effort. Addison-Wesley, Cambridge, Massachusetts.