Adaptive Approximate Record Matching

Ramin Rahnamoun[1]

[1] Computer Engineering Department, Azad University-Tehran Central Branch, Tehran, Iran. Email: r.rahnamoun@iauctb.ac.ir

Abstract

Typographical data entry errors and incomplete documents, produce imperfect records in real world databases. These errors generate distinct records which belong to the same entity. The aim of Approximate Record Matching is to find multiple records which belong to an entity. In this paper, an algorithm for Approximate Record Matching is proposed that can be adapted automatically with input error patterns. In field matching phase, edit distance method is used. Naturally, it had been customized for Persian language problems such as similarity of Persian characters, usual typographical errors in Persian, etc. In record matching phase, the importance of each field can be determined by specifying a coefficient related to each field. Coefficient of each field must be dynamically changed, because of changes of typographical error patterns. For this reason, Genetic Algorithm (GA) is used for supervised learning of coefficient values. The simulation results show the high abilities of this algorithm compared with other methods (such as Decision Trees).

Keywords: record matching, edit distance, data cleaning, genetic algorithms

1. Introduction

In real world, erroneous data entry is a recurring problem. System user typographical error is the major reason for this problem. Moreover, missing data in input documents and careless data entry increase error rate. As a result of these errors, multiple records would be saved in database tables which are similar to each other. In other words, an entity can be recorded in multiple forms. According to the case study of this research which is an insurance identification system, this process leads to creation of multiple records for an individual.

The approximate record matching algorithms look for similar records and assign them to a specific entity. There are two major phases for record matching process, which are search and matching phases. The first phase finds similar records in a database table. The second phase (matching phase) takes two records as inputs and if there is similarity between them, returns the similarity value. In this paper, the focus is on the matching phase. The main idea used for field matching is edit distance. The experiments in this research show that the algorithm must be customized for special problems in Persian language. Some of these problems are: similarity between Persian characters, the usual location of typographical errors in a string, etc.

In evaluating the similarity between records, each field has different levels of importance. The importance level of each field can be specified by a coefficient value. The specification of the coefficient values for each field can be evaluated experimentally. This paper uses a supervised learning algorithm (GA) to specify the coefficients. However, there are other automated methods for specifying the importance of each field in a record (e.g. Decision Trees) [9], it is possible that the input error pattern can be changed. It means that users may change their error pattern during the time. It is very important to notice that the main reason for errors in the system is carelessness of the users during the data entry.

This paper is organized as follows. In section 2 the record matching problem is defined. Then in section 3 edit distance concept is discussed. After that, in section 4 our case study is defined. In section 5 the proposed algorithm is discussed and then in section 6 the experimental results are represented. Finally, the conclusion remarks are made in section 7.

2. Edit Distance, Basic Definitions and Extensions

In real world, we must work with inconsistent, noisy and incomplete data [2]. So oftentimes it is the case that many records in a database refer to one real world object or entity. Record matching is the process of finding such records. Record matching needs some steps to be completed. First of all, data must be prepared for record matching. This step consists of some works such as selection of fields from table(s) for record matching, deletion of obvious noise from row data, etc. Generally, record matching consists of two major phases: searching and matching.

Searching phase involves finding the potentially linkable pairs of records. If we compare each record with all the other records, this phase will be so simple. But the time complexity of such search is $O(n^2)$ and in a large scale database it cannot be feasible [7]. So many different methods suggest for reducing searching time. Nested-Loop join, blocking methods (Soundex code), sorted neighbourhood approach and priority queue algorithms suggested for this phase [4] [6]. In this paper we focused on matching phase and therefore don't expand the mechanism of these methods.

Matching phase decides whether or not a given record pair is correctly matched. Typographical errors change two similar strings to different forms. So we need an approximation method for comparing these two strings and match them approximately. Matching phase can be divided to two steps. Field matching tries to match specific fields in record pairs. When pairs of fields are compared to each other and their distance computed, we must evaluate total distance between two records. Every selected field gets a weight based on importance of that field. Record matching methodology concentrate on assigning these weights to the fields.

Many different techniques have been applied for field matching such as: [3] [10]

Edit (Levenshtein) Distance: Is a measure of similarity between two strings. The distance is the number of insertion, deletion or substitution required to transform one string into the other.

Smith-Waterman: Given two strings, this method uses dynamic programming to evaluate minimum number of changes for transferring one string into the other. Smith-Waterman algorithm is resembled to edit distance but improve by a similarity matrix of alphabets.

N-Grams: n-gram is a vector representation that includes all the n-letter combination in a string. The string comparison algorithm forms n-grams vector for the two input strings and subtracts one vector from the other. The magnitude of the resulting vector difference is compared to a threshold value. This value is determined experimentally.

Recursive field matching algorithm: this method takes into account the recursive structure of typical textual fields [8]. If two strings are resembled or one abbreviates the other, they are matched with score 1.0. Each subfield of a string is assumed to correspond to the subfield of the other string with which it has the highest score.

3. Edit Distance, Basic Definitions and Extensions

My approach to finding distances of two strings is based on edit distance concept. So in this section I define basic concepts of this category and in next section try to describe further modification for my specific application.

The distance d(x, y) between two strings x and y is the minimal cost of operations that transform x into y. In edit distance, these operations are insertion, deletion and replacement.

Edit distance use dynamic programming technique. According to this technique, we must define a table and a method to fill it. So C is a matrix (table) that stores the cost of conversion. C_{ij} represents the minimum number of operations needed to match $x_{1..i}$ to $y_{1..j}$. Equation (1) calculates C_{ij} value:

$$C_{i,j} = \begin{cases} i & if\ j = 0 \\ j & if\ i = 0 \\ \begin{cases} C_{i-1,j-1} & if\ x_i = x_j \\ 1 + Min(C_{i-1,j}, C_{i,j-1}, C_{i-1,j-1}) \end{cases} \end{cases} \quad (1)$$

Smith-Waterman algorithm is an extension to basic edit distance idea. This algorithm is used for DNA analysis. But some of researcher used it for string processing. The Smith-Waterman algorithm uses a matrix of similarity between alphabets. So two symbol of alphabet may be matched, approximate matched or mismatched. The degree of matching between two symbols related to the value of similarity of them inside similarity matrix. Much of the power of this algorithm is due to its ability to introduce gaps in the fields of records. But in this application, gaps inside the fields are not produced.

4. Case Study

Identifying persons that benefit from their services is one of the major tasks in a social security organization. So this organization needs an identification system. Social Security Organization of Iran (SSOI) has more than 400 branches around the country. Each branch of SSOI has an independent identification system. For this reason, one person may be identified in different branches and gets multiple Social Security Numbers (SSN).Moreover, in a specific branch, careless data entry, typographical errors, insufficient data in input documents and other

factors, make multiple database records that belong to unique person.

Now, SSOI try to design a centralized identification system for all the branches around the country. So data cleaning process for generating identification database will be a major task. Table 1 shows data entry error rates in a large scale branch of SSOI. It can be inferred from this table that careless data entry is a serious problem.

Table.1
The statistics related to a branch of SSOI. Total number of records is 673914. This statistics show obvious data entry errors. For example, special characters in name and family fields, unusual short length fields (one character in family field), etc.

Identity field name	Error percent
Family	0.43
Name	0.55
Father's name	8.94
ID	4.8
Birth date	6.2
Birth place	12.1

5. Proposed Algorithm for Approximate Record Matching

My proposed algorithm is based on edit distance concept. This algorithm can be divided into two distinct phases: Field Matching and Record Matching. In the first phase, the algorithm concentrate on similarity of two fields from two different records, but in the second phase record matching and the importance of fields is considered. Also we need a preprocess phase to reduce noise of data.

5.1. Preprocess phase

Incorrect data in each field may be entered on purpose or not. SSOI Software application cannot verify data entry quality with high precision, so we find special characters such as "?" or" –"in name or family fields (refer to table 1). Finding and eliminating such erroneous data can improve the accuracy of the algorithm.

Otherwise, this study indicates a specific error pattern. The repetition of a specific column's value in a table is an anomalous manner. Figure 1 shows the histogram of birth date filed in our case study. This chart shows that a jump in 1347 (this year belongs to solar calendar). Evidently, this is an abnormal case in this field and must be identified and removed from the table. In this paper, just evident samples of this case study (such as special characters in name and family fields) were deleted. It seems this problem needs an independent research.

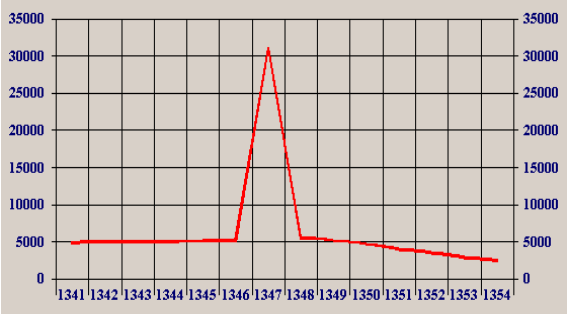

Fig.1. The histogram of birth date (horizontal axis) versus density (vertical axis). An abnormal jump is detected in 1347. All years are in solar calendar.

5.2. Field Matching Phase

The Filed matching phase in this specific application has some important key points. If the algorithm does not concern on these points, performance of it decrease drastically. But what are these key points?

One of the important points of the Smith-Waterman algorithm is to specify the similarity matrix. So the algorithm must determine this matrix between Persian characters. In the first step, a corpus of typographical errors in Persian language was formed. Based on these empirical results, characters with most typographical errors are selected as most similar characters. Table 2 is an example character similarities.

Table.2
Important rows of similarity matrix for Persian characters. The total number of samples is 9011.

	Percent of similarity	First character	Second character
	0.9	ت	ا
	1.6	ن	ت
	1.4	ح	ج
	2.1	خ	ج
	1.4	خ	ح
	5.7	ز	ر
	3.7	ش	س
	0.9	ض	ص
	2.4	غ	ع
	7.5	ن	م

Another problem is the location of typographical errors in an input string. This research focused on name and family fields. This study shows that the probability of error increases near the end of input string. So, leading input characters have important role. Reformulated relation 1 for this point is:

$$C_{i,j} = \begin{cases} i & if\ j = 0 \\ j & if\ i = 0 \\ \begin{cases} C_{i-1,j-1} & if\ x_i = x_j \\ 1 + Min(C_{i-1,j} + (1 - \frac{i}{len} \times slope), & \\ C_{i,j-1} + (1 - \frac{i}{len} \times slope), C_{i-1,j-1}) & \end{cases} \end{cases} \quad (2)$$

In this relation, "len" is the length of each string, "slope" is a coefficient and "i" is the index of character in input string.

5.3. Record Matching Phase

In this phase, contents of each field in two records must be compared. If S_{ijk} shows the similarity between kth field of two records (i, j), then $\sum_{k \in R} S_{ijk}$ is the approximate record matching value. In this relation, R is the domain of fields of each record. It is obvious that total sum of these values cannot produce the proper result. For example, in this case study (identity records), most of data in birth date field were incorrectly entered. The application forced the operator to enter this value, but there is no value in these input documents. So the operator entered the incorrect values. Therefore in our calculations, the birth date field has less importance.

We must determine a coefficient for each field to find the importance of it. This is computed as:

$$Sim(i, j) = \frac{\sum_{k \in R} \left[1 - \frac{d_{ijk}}{Max\{Len_{ik}, Len_{jk}\}} \right] B_k}{\sum_{k \in R} B_k} \quad (3)$$

In this relation:
Sim(i, j) : the similarity of two records i, j.
d_{ijk} : extended edit distance of kth field of two records i, j.
Len_{ik} : length of kth field from record i.
B_k : the coefficient of field k.

Finding the B_k values is a complex problem, so brute force method cannot solve it. We need a machine learning method to find optimal B_k values. In this paper, a simple GA used to find optimal values of the coefficients [1].

In this approach, each gene consists of a five tuple that each member of it is a coefficient. Mutation and crossover operators are used same as classical forms. Fitness function is based on relation 3. For this purpose, a training set prepared (refer to section 6) and the suggested coefficients feed to relation 3.

6. Experimental Results

Supervised learning needs training set that its similar and non-similar records must be identified and also needs a test set to evaluate the suggested algorithms. Similar records in training set were produced by using a specific file in application software. This file is the history of typographical error correction by operators. But for non-similar samples, I choose two workshops with different activities and geographical locations. Test set was generated by a similar manner.

The total number of records in training set is 761. Table 3 indicates the results of GA on the training set.

This paper used SQL server analyser manager software for implementing Decision Trees (DT). Fig.2 represent the suggested DT of this software.

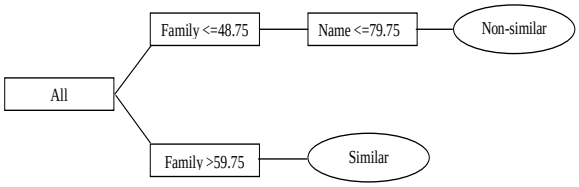

Fig.2. Decision Tree, generated based on training set.

The two algorithms (GA, DT) trained by this sample and after that they were tested. The abilities of these algorithms are showed in table 4. The results of this table represent that they have similar discriminating power.

Table.3
Simulation results based on GA. All of the values in table are in percent (except for fitness and population size).

Population Size	name	family	father name	ID No.	birth date	fitness
50	36	55	2	5	2	0.0302
50	38	56	0	6	0	0.0272
100	40	54	0	6	0	0.025
100	40	53	0	7	0	0.025

Table.4
The comparison between GA and DT.

	GA	DT
Accuracy percentage	97.3	96.8

7. Conclusions

Real world databases have considerable errors in their data. Data cleaning [2] [5] has various approaches that each of them consisting of many phases. This paper focuses on matching phase in approximation record matching.

Edit distance is the basic method of field matching phase. The experimental results show that attention to specific problems of each application (SSOI in this case) can improve the discrimination capability of the algorithm. So, this research concerns

on specific problems such as similarity of Persian characters, custom location of errors.

Otherwise, in record matching phase each field has a coefficient and in this way the importance of each field in a record can be determined. For determining these coefficients, we use GA as a supervised learning method. The experimental results based on test set show that this algorithm is successful. But in other research, decision trees were used for this problem and decision rules were produced. So we compare these two algorithms and find their similar discriminating abilities. In this case study, the operators repeated pattern errors seriously. These patterns may be changed intelligently during time. So, the flexibility of matching algorithm during time is important. In GA, we tackle this problem by feeding new training samples into system and then correct the coefficients. But decision trees cannot handle this problem easily.

The last point is the importance of three fields: family, name and ID. Both methods find the two other fields (birth date and father name) are worthless.

References

[1] D. E. Goldberg, “Genetic Algorithms in Search Optimization and Machine Learning”, Addison_Wesley, 1989.

[2]] J. Han, M. Kamber, “Data Mining: Concepts and Techniques”, Morgan Kaufmann, 2001.

[3] M. A. Hernadez, S.J.Stolfo, "Real-world Data is Dirty: Data Cleansing and the Merge/Purge Problem", Journal of Data Mining and Knowledge Discovery, Vol.1, No.2, 1998

[4] J.A. Hylton, “Identifying and Merging Related Bibliographical Records”, Master's Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1996.

[5] M. Kantardzic,” Data Mining: Concepts, Methods, and Algorithms”, IEEE Press, 2003.

[6] K. Kukich, "Techniques for Automatically Correcting Words in Text", ACM Computing Survey, Vol.24, No.4, 1992.

[7] A. E. Monge, “Adaptive Detection of Approximately Duplicate Database Records and Database Integration Approach to Information Discovery”, PHD Thesis, University of California, San Diego, 1997.

[8]] A. E. Monge, C. P. Elkan, "The Field Matching Problem: Algorithms and Applications" Second International Conference of Knowledge Discovery and Data Mining, AAAI Press, 1996.

[9] V. S. Verykios, A.K.Elmagarmid, E.H.Houstis, "Automating the Approximate Record Matching Process", Information Science, Vol.126, No 1-4, 2000.

[10] V. S. Verykios, G.V.Moustakides, "A Cost Optimal Decision Model for Record Matching", Workshop on Data Quality, 2001.