Volume: 6 Issue: 3 Pages: 2334-2337 (2014) ISSN : 0975-0290

String Identifier in Multiple Medical Databases

Simona-Roxana Dumitrescu
Automatic Control and Systems Engineering Department, Univerity Polytechnics, Bucharest, Romania
Email: simona_roxana_robeci@yahoo.com
Dan Popescu
Automatic Control and Systems Engineering Department, Univerity Polytechnics, Bucharest, Romania
Email: dan_popescu_2002@yahoo.com

----------ABSTRACT----------
In a distributed medical system, building cross-site records while maintaining appropriate patients anonymity is essential. The distributed databases contain information about the same individuals, often described by using the same variables, which do not fit quite frequently due to accidental distortions. In such cases, the record linkage methods are used to find records that correspond to the same individuals in order to create a consistent database. Our goal was to find a solution for this problem. In this paper, we propose an anonymous identifier, based on combinations of first two letters from the surname, name, date of birth and gender, which can allow a de-identifying merged dataset from multiple databases of a distributed medical system.

Keywords - **record linkage, identifier, matching algorithm, Jaro-Winkler.**

Paper submitted: 15th August 2014, Revised: Date (only if applicable), Accepted: 5th November 2014

I. INTRODUCTION

An integrated medical system contains multiple distributed databases which are different from the content and design point of view. Therefore, the same patient may be registered in one or more databases, depending on his medical records and history. Often, duplicate records don't use the same identifier and may contain erroneous data, which makes the matching process to be extremely difficult [1,2,3].

The process becomes even more challenging when the requirement of data privacy has to be assured. Gathering consistent medical information about one patient involves a systematic process for comparing the identifiers. This task is based on data mining methods and records linkage. Therefore it should be validated and should satisfy all the national constraints regarding the medical data [3].

The record linkage algorithm should also include the verification of appearance rate of duplicated data. The selection of a unique identifier is a delicate issue due to security policies of each country [3,4,5]. However, in our country, the patient identification is made based on the CNP (personal numeric number).

Our goal was to find a solution for this problem. In this paper we propose an anonymous identifier, which can allow a de-identifying merged dataset from multiple databases of a distributed medical system. The identifier is a string obtained by combination of first two letters from the name, surname, date of birth and gender. This article presents in details how we created and tested the identifier referred in [6]. Our algorithm was based on the idea that the most inadvertent variation in names occurs after the first two letters. Taking into account that nor the name, neither the date of birth individually cannot be considered as a valid identifier, we considered that by combining these we can obtain an identifier with the desired features. Also, considering that the databases contains records both genders records, we included the gender fields in order to obtain a strong identifier.

II. METHOD USED FOR TESTING THE IDENTIFIER

In order to test our solution, we used the Jaro-Winkler method to identify records that describe the same person.

The record linkage is a preprocessing technique used for data cleaning and data integration in distributed and heterogeneous databases. The quality of a matching records system heavily depends if the chosen approach allows accurate detection of duplicates in an effective and efficient way [7].

Numerous methods have been proposed for record linkage. These methods are classified in three major categories: distance-based methods, token-based methods and phonetics based methods. One of the most currently used distance-based method is Jaro-Winkler distance [8], a strong method for comparing the similarities of short strings. This was one of the main reasons we choose to test our solution using this metric. The method represents the calculation of a score based on the sum of the characters which perfectly match. The distance Jaro–Winkler is used for comparing short strings, such as names. The score is normalized so that 0 means no similarities and 1 represents

Volume: 6 Issue: 3 Pages: 2334-2337 (2014) ISSN : 0975-0290

a perfect matching. The Jaro–Winkler similarity between strings x and y is defined as (1):

Jaro-Winkler(x,y) = Jaro(x,y) + 0,1×max{4,LCP(x,y)}×(1-Jaro(x,y)) (1)

where Jaro(x, y) is the Jaro similarity, and LCP(x, y) is the length of common prefix at the start of the string up to a maximum of 4 characters. The most important step of the data linkage is an n-to-n systematic comparison between a patient and the whole distributed datasets of already integrated databases. In order to avoid the case when the linkage process will produce an unacceptable proportion of false positives, in the matching algorithm one have to define two thresholds (T1 and T2) [3].

The creation of two thresholds will define a new area between two automatic decisions (true positive and true negative). The new defined area needs a manual intervention to determine if the patient matches or not.

As described in figure 1, the matching algorithm aggregates the identities if the score is greater than a maximum threshold T2 (already known patient).

In the case when the score is lower than the minimum threshold T1, it creates a new patient with a new identity (unknown patient). If the score is between the minimum and the maximum thresholds, the automatic linkage can introduce errors and a manual intervention is needed.

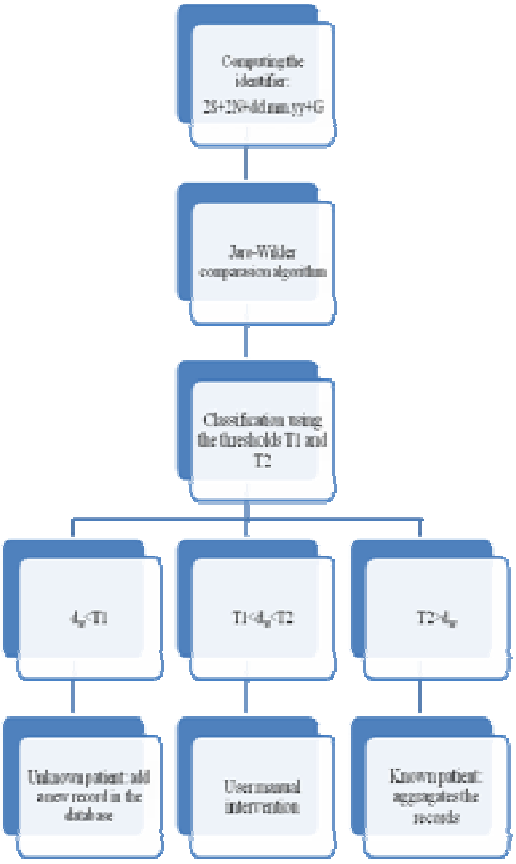

Figure1. Matching process diagram

The Matlab code of the algorithm used in the tests is the following (s_1 and s_2 are the strings compared, n is the length of the common prefix for four characters, $common_{12}$ and $common_{21}$ represent the total number of common characters and p is the scale factor) [6].

```
function [dj,dw]=getJaroWinkler(s1,s2,dpmin,p)

dmax=floor(max(length(s1), length(s2))/2)-1;  % get the distance over which we match characters

%get common characters in both directions
common12=getCommon(s1,s2,dmax)
common21=getCommon(s2,s1,dmax)

if ~isempty(common12) && ~isempty(common21)
    dmin=min(length(common12),length(common21));
    transpositions=sum(common12(1:dmin)~=common21(
    1:dmin))/2
else
   warning 'at least of the common matching strings was empty'
   transpositions =0;
end
if length(common12)> 0
   dj=(length(common12)/length(s1)+length(common21)/
length(s2)+(length(common12)-transpositions)/length
(common12))/3;
else
   dj=0;
end
n=getPrefixLength(s1,s2,dpmin);
dw=dj+n*p*(1-dj);
```

The function that calculates the length of the first maximum of four characters common is [6]:

```
function n=getPrefixLength(s1,s2,dpmin)

n=min([dpmin length(s1) length(s2)])
ndiff=find(s1(1:n)~=s2(1:n))
if ~isempty(ndiff)
   n=ndiff(1)-1
   n=n-length(ndiff)
else
n=min([dpmin length(s1) length(s2)])
end
```

The function that calculates the total number of common characters is [6]:

```
function [common]=getCommon(s1,s2,dmax)

common='';
for i=1:length(s1)
   tmp=strfind(s2(max(1,i-
dmax+1):min(length(s2),i+dmax)),s1(i));
```

Volume: 6 Issue: 3 Pages: 2334-2337 (2014) ISSN : 0975-0290

```
if ~isempty(tmp)
    common=[common s1(i)];
end
end
```

III. DESIGNING THE IDENTIFIER

In order to obtain proper and accurate results it is very important how we choose the succession of the characters of the identifier. According to its definition, the length of the common prefix at the start of the strings up to a maximum of 4 characters has the higher weight in computing the Jaro-Winkler distance [8,9,10].
Following the tests performed, we have observed that if we choose to compute the identifier based on calculation of the first 2 letters from surname, name, date of birth and gender (taken in this succession) the matching algorithm will generate false results, because the probability of finding two persons which first two letters of the name and surname will match is higher.

For example, if we want to compare the identifier of a new entry ID =VADA180485F (name =VASILESCU, surname= DANA, date of birth=18.04.1985, gender =F) with an already existing record with ID=VADA180485M (name=VASILE, surname= DANIEL, date of birth =18.04.1985, gender = M), the Jaro-Winkler distance will be dw= 0,9636; this result is very close to 1 and the matching algorithm will consider that the two strings matches.

Therefore, we have reconsidered the succession of characters as follow: gender, date of birth, first two letters of surname and name. Computing the Jaro-Winkler score for the example above, but changing the ID, we have obtained dw= 0.9576, which is lower that the first result. In both situations the two strings differs only by one character. But this character indicates the gender, so the two strings don't match. That's why, the second option conveys more to our goal. Of course, during the matching process, the value 0.9576 will not be included in the true positive interval.

IV. EXPERIMENTAL RESULTS

In order to define T1 and T2 used in matching algorithm, we performed a lot of tests covering all the possible cases met in practice.

We started to evaluate the possible combinations of the common characters of the given strings taken into account the formula used to calculate the Jaro-Winkler distance. According to its definition, the length of the common prefix at the start of the string up to a maximum of 4 characters has the higher weight in computing the Jaro-Winkler distance. Also, the number of common characters of the compared strings has a high weight in calculating this score [7].

Considering l- length of the common prefix at the start of the strings up to a maximum of 4 characters and commons1s2 - number of common characters in the interval [character 5, character 11] we have obtained the results in the tables below.

Table 1.For l=0 and common $s_1s_2 \in [0,7]$

l=0/Common s_1s_2	**d_w**
0	0
1	0.3939
2	0.4545
3	0.5152
4	0.5758
5	0.6364
6	0.6970
7	0.7576

Table 2.For l=1 and common $s_1s_2 \in [1,8]$

l=1/Commons$_1$s$_2$	**d_w**
1	0.4545
2	0.5091
3	0.5636
4	0.6182
5	0.6727
6	0.7273
7	0.7818
8	0.8364

Table 3.For l=2 and common $s_1s_2 \in [2,9]$

l=2/ Common s_1s_2	**d_w**
2	0.5636
3	0.6121
4	0.6606
5	0.7091
6	0.7576
7	0.8061
8	0.8545
9	0.9030

Table 4.For l=3 and common $s_1s_2 \in [3,10]$

l=3/common s_1s_2	**d_w**
3	0.6606
4	0.7030
5	0.7455
6	0.7879
7	0.8303
8	0.8727
9	0.9152
10	0.9576

Volume: 6 Issue: 3 Pages: 2334-2337 (2014) ISSN : 0975-0290

Table 5.For l=4 and $s_1s_2 \in$ [4,11]

l=4/common s_1s_2	d_w
4	0.7455
5	0.7818
6	0.8182
7	0.8545
8	0.8909
9	0.9273
10	0.9636
11	1

V. CONCLUSION

We have determined the values for optimal thresholds T1 and T2 based on data obtained from conducted tests. Therefore, in order to achieve the best results and to eliminate the situations when the algorithm may introduce erroneous data, we have considered T1=0,800 and T2=0,960.

Higher values than T2 corresponds to the situations of true positive. In this case, the matching algorithm founds records whose identifiers coincide with the compared one. For values lower than T1, no records whose identifier corresponds to the one compared were found. In this situation (the negative case), the unknown record will be added in the database. For the results obtained in the interval [T1,T2] an automatic data linkage might introduce erroneous results. Thus, a manual intervention is required.

The main problem that we have encountered in our tests was that until now we couldn't validate our solution on a real database, due to the access databases restrictions.

The research was conducted on a test database with o low number of records (about 300). So, practically the possibility of finding situations of false positive and false negative tends to 0 and we have obtained 100% sensitivity and 100% sensibility of the matching algorithm.

We are currently continue the research in order to test our solution on a dataset much higher that the first one. Using a unique identifier and data linkage techniques, the process of patient's identification in distributed databases can be done in an efficient manner.

As a future development we intend to test our solution with other record linkage method, the Levenshtein algorithm, and also to compare our results with the results obtained evaluating others combinations of identifiers in terms of their sensitivity and specificity.

ACKNOWLEDGEMENTS

This work was supported in part by the Romanian Ministry of Education and Research under grant of POSDRU Project ID 76903.

REFERENCES

[1] Tsung-Chih Hsiao, Zhen-Yu Wu, Yu-Fang Chung, "A secure integrated medical information system", Journal of Medical Systems, 36(5), pp. 3103-3113, Oct. 2012

[2] Steven Walczak, "A multiagent architecture for developping medical information retrieval agents", Journal of Medical Systems, 27(5), pp.479-498, Oct. 2003

[3] Sebastian Cipiere, Paul De Vlieger, "Development of a metamodel for a medical database management on a grid network", ACM International Symposium on Cluster, Cloud and Grid Computing, 2012

[4] Durham E, Xue Y, Kantarcioglu M and Malin B., Private Medical Record Linkage with Aproximate Matching, AMIA Annnu Symp Proc., 2010:182-186

[5] Kijsanayotin B, Speedie S, Connelly D, Linking Patient's Records Across Organisation While Maintaining Anonymity, AMIA Annun Symp Proc, 2007 Oct 11:1008

[6] Simona Dumitrescu, Ioana BrănescuRăspop, Dan Popescu, Radu Dobrescu, Format medi- ation for matching patient records in a three-layer Information Architecture,International Symposium on Electrical and Electronics Engineering (ISEEE),October 2013, Galați, Romania

[7] Arellano MG, Weber Gl, Issue in identification and linkage of patient records across an integrated delivery system, J Health Inf Manag Fall 1998; 12(3):43-52

[8] Winkler, W.E. 1999. The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service Publication R99/04. Available from http://www.census.gov/srd/www/byname.html

[9] Fellegi, I. P., and A. B. Sunter (1969), "A Theory for Record Linkage," Journal of the American Statistical Association, 64, pp. 1183-1210

[10] Winkler, William E. "Matching and Record Linkage". U.S. Bureau of the Census. Retrieved 12 November 2011 at http://www.census.gov/srd/papers/pdf/rr93-8.pdf