Improving the Efficacy of Approximate Searching by Personal-Name

Rafael Camps, Jordi Daudé

Software Department, Universitat Politècnica de Catalunya,
C/ Jordi Girona 1-3, 08034 Barcelona, Spain
rcamps@lsi.upc.es

Abstract: We discuss the design and evaluation of a method to find the information of a person, using his/her name as a search key, even if it has deformations. We present a similarity function that is an edit distance function with costs based on the probabilities of the edit operations but depending on the involved letters and their position. The distance threshold varies with the length of the searched name. The evaluation of the efficacy of approximate matching methods is usually done by subjective relevance judgements. An objective comparison of five methods, reveals that the proposed function highly improves the efficacy: for a *recall* of 94%, a *fallout* of 0.2% is obtained.

1. Introduction

Data related to people are stored in almost all Information Systems (IS): customers, patients, taxpayers, etc. In the databases (DB) of IS, the percentage of people with errors in their names is often close to 30%. Frequently, the sources of data are very diverse and therefore the causes of deformation can be numerous.

The aim of this work is to present a high efficacy method to determine if two names are similar. We consider that two names are similar, if with a certain probability both refer to the same person. For example, in the pair *VELASCO / BLASCO* very probably one of them is consequence of deformations in the oral transmission or the writing of the other, but it is not probable that this is the case of the pair *VELASCO / MARTIN*. The method is devised to find people in a DB, using the personal-name as search key, in spite of errors. But it also can be used in the *Named Entity Recognition* problem [TJ02], where the names appear within a free text.

We can express our problem as follows: Given a name x and a set of names C_1, obtain a set $C_2 \subseteq C_1$ with those names that are similar to x. The C_2 set may be empty. We can introduce a parameter k to tune the similarity criterion. Expressed in form of a function, we have

$$C_2 = Similars\ (x, C_1, k)$$

This function is usually based on a *distance* function $\delta(x, y)$ that, given two names, determines a measurement of its dissimilarity. Then, k will be the threshold of that distance. In our case, the C_1 set is a vocabulary of names existing in the DB in which we want to search. We intend to relate the value of this distance with the probability that *x* does not refer to the same person than *y*.

Distance functions [VE88], [BO92], [KU92], [PF96] and [ZO96] are more effective than phonetic codification methods [KU92], especially because codification produces more false positives than distance functions. Nevertheless, distance functions have the disadvantage that the associated search techniques are usually very time consuming (very inefficient). For the DEA distance function that we present in the next section, it is possible to use an efficient approximate-searching technique based on a trie-tree. The efficacy of DEA, evaluated objectively (section 3) is very high: for example, for a *recall* of 94%, a *fallout* of 0.2% is obtained.

Our work is oriented to the Spanish context, and it is mainly centered in surnames. First names are usually submitted to a very different treatment than surnames. Anomalies of the macrostructure, such as transposition between the parts of a name, are not contemplated here, but they are well studied in [FR97]. Our research is described in more detail in [CA03]. From now on we will use the terms *name* and *surname* indistinctly.

2. The DEA Distance Function

2.1 Distance Functions

The most popular dissimilarity measure between two character strings, is the simple edit distance or *simple distance* for short [NA01] defined as the minimum number of edit operations, insert (`I`), delete (`D`) and substitution (`S`), needed to transform one string into the other. In a weighted distance function the three edit operations can have different costs depending on the characters. For example, a substitution of an *M* by an *N* can have a cost lower than a substitution of *M* by *R*. The edit distance $\delta(x,y)$, can be defined as the minimum cost of all the possible sequences that transform *x* into *y*. The cost of a sequence of operations is the sum of their costs. In simple distance all the costs are equal to 1.

Here we will propose a distance function: DEA. It is an edit distance for which we define a variable threshold depending on the length of the searched name, and with operation costs according to a probabilistic model that tries to catch the deformations that actually occur in a corpus, whatever the causes are. The operation costs will depend on; the type of operation (`I,D` or `S`), the position where the operation is applied, and the letters involved in the operation.

The calculation of a distance usually assumes a previous transformation of the characters in order to obtain a normalized string format that depends on the application. We apply to the personal-names a normalization process that basically consists in turn lowercases to uppercases and deletion of diacritics and other symbols than letters or blanks.

2.2 Discrimination and Cost Estimation

Our approach to estimate the costs of the elementary edit operations, is based on the discrimination concept. Let us call *pairs-with-error* a set of pairs of similar names (one pair member is an erroneous version of the other) and *pairs-without-error* a set of pairs of independent names (each one refers to a different person). We call *discrimination* of an edit operation, the ratio between the probability of its occurrence in the set of *pairs-without-error* and the probability of occurrence in the set of *pairs-with-error*. We will use as *pairs-with-error* corpus, a *TEST* file containing 10593 real cases of pairs of surnames, in such way that one surname is an error or deformation of the other. As *pairs-with-error* corpus, we will use a *CONTROL* file containing 9345 pairs, obtained randomly pairing surnames.

The discrimination D_{op} of an edit operation (*op*) is given by

$$D_{op} = \frac{Pr(\text{op in } CONTROL)}{Pr\ (\text{op in } TEST)}$$

In order to be able to use an efficient search strategy (for example a pruning technique) we need that the cost of each operation is not greater than the cost of an equivalent sequence of operations, that is, we need that the distance satisfies the triangular inequality. Therefore, in order to use the discriminations as costs, we scale them in such a way that *Dmin = Dmax/2.*

To improve the efficacy we also take into account the position where the edit operation occur. We distinguish between the *first* position, the *last* position and the other positions, or *general* position. The probabilistic model actually used, consists of three confusion matrix with the probabilities of the 1053 = $3*(26+((26^2-26)/2))$ different operations (for the numerator of the discrimination) and three vectors of prior probabilities (for the denominator). To obtain the 1053 costs, we transform the discriminations, in such a way to comply the triangular inequality.

2.3 Thresholds

The number of errors made is not independent of the length of the name. Therefore, the parameter k , the distance threshold to choose the similarity degree, in the function *Similars* (x , C_1 , k), should depend on the length of the name. In order to facilitate the comparison of DEA with other functions, we have decided to use seven degrees of similarity (*A,B,C,D,E,F and G*). For each degree several threshold values are needed, one for each query length.

3. Evaluating and Comparing Distance Functions

3.1 Other Distance Functions

Through the years, large amounts of proposals have been made for the determination of the similarity of two words, based on the comparison of its characters. We will empirically compare the DEA distance with the simple distance and with the three following distances;

Bigrams: The distance expression used by us is:

$$\delta(x,y) = \frac{Bx + By - 2Bxy}{2Bxy}$$

were B_x is the number of different bigrams existing in the word x , B_y is the number of different bigrams existing in y, and B_{xy} is the number of different bigrams common to both words. When there are no common bigrams, the value of B_{xy} will be 0.5.

Jaro: A distance devised specifically for surnames, used to detect coincidences of people during the processes of the US Census [WI95].

Editex: Zobel and Dart [ZO96] proposed this comparison function, that they tested using surnames. Editex is a variant of a weighted edit distance, where only three different costs exist: coincidence, similar and non-similar. The similarity criterion is based on the phonetic groups of the PHONIX codification system [GA90].

3.2 Subjective and Objective Evaluations

The efficacy is related to the hits and faults in the identification of the similarity. In the area of Information Retrieval (IR), *relevance judgements* made by human judges are used to decide if the retrieved documents are relevant or not to the query. When searching people by name, this type of evaluation procedure is not appropriate because it is too subjective and unsteady. Even though, it is the procedure traditionally employed in the comparison of name matching methods [BO92], [PF96], [ZO96] and [PE00].

In order to avoid the evaluation subjectivity, we adopt an approach based on a corpus containing real deformations. We are interested in the empirical evaluation of how the different distance functions are able to correctly discriminate between *pairs-with-error* (a test file) and *pairs-without-error* (a control file). We will not use the same files (*TEST / CONTROL*) that were used in section 2 for the determination of the DEA costs, but another pair of files *TESTR / CONTROLR*. Now suppose that we have all the surname pairs of both files, *TESTR* and *CONTROLR*, together into a single set, and we try to identify the pairs pertaining to *TESTR*, that is, the *pairs-with-error*.

If we use a distance threshold as a discrimination criterion, a partition is produced in four sets: a) pairs-with-error (pairs from *TESTR*) identified correctly as such, b) ***false positives*** pairs, that is pairs-without-error (pairs actually from *CONTROLR*) identified as pairs-with-error, c) ***false negatives***, that is pairs-with-error (pairs actually form *TESTR*) identified as pairs-without-error, d) pairs-without-error (pairs from *CONTROLR*) identified correctly as such. The total number, *N*, of pairs of the experiment, is the number of pairs in *TESTR* plus the number of pairs in *CONTROLR.*

The metrics we will use to quantify the two anomalies are *Fallout* and *Recall*:

- The *Fallout* is the probability that a pair-without-error is a false positive. It will be noted as *F*.
- The *Recall*, noted as *R*, is the probability that a pair-with-error is identified as such. Often we prefer to use its complement, named here with the term *misidentification*. The misidentification is the probability that a pair-with-error is a false negative. So, *R* = 1- misidentification.

In table 1, the values of *F* and *1- R* are given for the five distance functions that we are analyzing, and for several thresholds.

Table 1: Evaluation of the distance functions (values in %)

Function	DEA			JARO		BIGRAM	EDITEX	SimpleDis
Threshold	C	D	E	0.14	0.18	1.1	6	2
Missid. 1-*R*	7.9	5.9	3.8	13.29	6.55	11.94	8.67	11.17
Fallout *F*	0.005	0.19	0.77	0.19	0.77	0.19	0.38	0.19
Prec. *P* β=1	99.99	99.79	99.20	99.78	99.18	99.78	99.58	99.78
“ *P* $\beta=10^{-3}$	94.85	33.12	11.11	31.33	10.82	31.67	19.38	31.85

3.3 The *MiFa* Graphic

In figure 1a we display the relationship between *1-R* and *F* for our corpus: *TESTR* and *CONTROLR*. We have called *MiFa* the graphic that relates the misidentification with the fallout. This graphic allows choosing the more appropriate threshold for each application. In the IR field, sometimes a graphic *Recall/Fallout* is used, though the *Recall/Precision* graphic (see point 3.5) is more popular. The *MiFa* graphic is widely used (under other names) in other fields as for example in biometrics or clinical research.

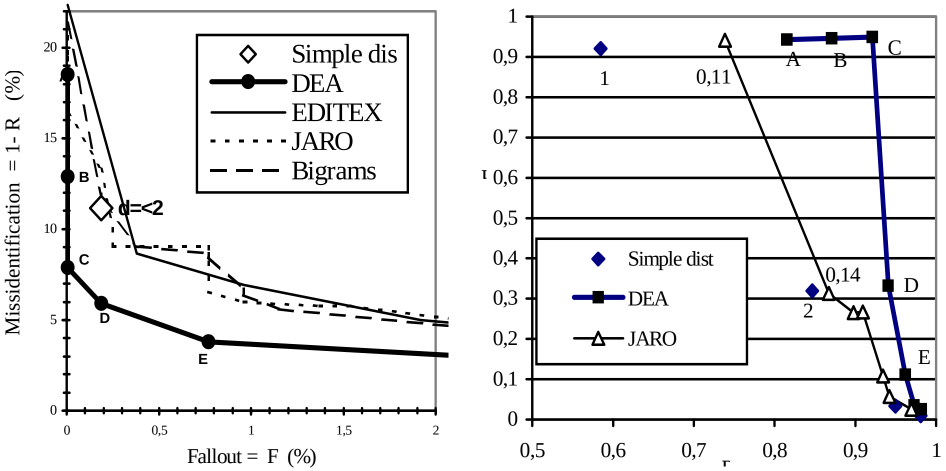
Usually, in practice, misidentification values greater than 22%, and fallout values greater than 2%, are totally unacceptable. Therefore, in figure 1a we limit the *F* and 1-*R* values to this interval. Into this interval, the simple distance function does not allow to tune the similarity criterion, the threshold, because only a single point exists ($\delta \leq 2$, since $\delta \leq 1$ and $\delta \leq 3$ are out of this interval and the simple distance values are integers). The DEA function has several points corresponding to the thresholds we have defined depending on the lengths (see section 2) but more points could be defined because DEA produces, within this interval, more than 100 different distances. The methods of phonetic codification are out of this interval; for a fallout of 1% they have a recall lower than 30%.

Our target is to minimize both, *F* and 1-*R*. A look at figure 1a, shows that DEA is the function that better fulfills our target. For the same level of misidentification (or recall), the DEA function gives a 70% to 80% lower fallout than the other functions (within our working interval). For the same level of fallout, it gives a 40% to 55% lower misidentification.

3.4 Data Volumes, β-Factor and Precision

With the values of *R* and *F* alone, it is not possible to compute the volumes of the four sets of the partition produced by a threshold (see 3.2). Therefore in order to predict these volumes, for example the number of false positives, we will also use *N*, the total number of pairs, and a factor β, expressing the ratio between the number of pairs-with-error and pairs-without-error. For most of applications involving personal-names, this β factor will have values lower than 0.01 and very often lower than 0.001 . Now we can express the number of non-desired answers (the false positives) as a function of *N*, *F* and β, by :

$$N\frac{F}{\beta+1}$$



a) Graphic *MiFa*

b) Recall/Precision graphic for β=0.001

Figure 1: Graphics for *TESTR/CONTROLR*

See that for high *N* values and low β (these are the usual conditions) the number of not-desired answers is very high, although the *F* value can look very low. It can happen that the misidentification and the fallout are both rather low, but the ratio of false positives, is very high. This can be unacceptable for many applications, because of psychological reasons or the difficulty of handling the answer. Therefore, it may be useful to use the *Precision*, *P*, metric, very common in IR. We can define the precision as the probability that a pair identified as a pair-with-error, is really a pair-with-error. So:

$$P=\frac{\beta R}{\beta R+F}$$

See table 1 for some precision values.

3.5 The *Recall/Precision* Graphic

For a given recall level, if the proportion β of names in the DB that are similar to the searched one, decreases, the precision decreases very quickly. In figure 1b we display for $\beta = 0.001$ the *Recall/Precision* graphic for the files *TESTR/CONTROLR*. The lines for the Bigrams and Editex functions are not displayed because their behavior is nearly the same than the Jaro function (see table 1). To obtain, with $\beta = 0.001$, a precision greater than 50%, we need to accept a recall *R* lower than 95%. To obtain a precision greater than 50% using the simple distance, we need a threshold δ=1, but that produces a very low recall (58.56%).

4. Conclusions

The simple edit distance is not an appropriate function. With the threshold $\delta \leq 3$ too much fallout is produced. With the threshold $\delta \leq 1$ too many false negatives (too low recall) are obtained. The threshold $\delta \leq 2$ still produces more false negatives (recall lower than 89%) but in some circumstances it can be accepted. No intermediate thresholds are possible. The other functions compared in this work, except DEA, are not significantly better than the simple distance. But the DEA function gives us important improvements. For the same level of recall, the DEA function gives a fallout from 70% to 80% lower, and for the same level of fallout, it gives a misidentification from 40% to 55% lower.

References

[BO92] Borgman, C.L; Siegfried, S.L.: Getty's Synoname and its cousins: A survey of applications of personal name matching algorithms. JASIS (43,7). 1992; 459-476.

[CA03] Camps, R.; Daudé, J.: Searching by approximate personal-name matching. Report LSI-03-9-R. Universitat Politècnica de Catalunya, Barcelona, 2003.

[FR97] French J.C. et al.: Applications of Approximate Word Matching in Information Retrieval. CIKM 97. 1997; 9-15.

[GA90] Gadd, T.N.: PHONIX: the algorithm. Program (24, 4). 1990; 363-366.

[KU92] Kukich, K.: Techniques for Automatically Correcting Words in Text. ACM-CS (24,4). 1992; 377-439.

[NA01] Navarro, G.: A guided tour to approximate string matching. ACM-CS (33,1). 2001; 32-88.

[PE00] Petrakis, E.G.M; Tzeras,K.: Similarity Searching in the CORDIS Text Database. Software Practice and Experience (13). 2000; 1447-1464.

[PF96] Pfeifer, U. et al.: Retrieval effectiveness of proper name search methods. Information Processing and Management (32,6). 1996; 667-679.

[TJ02] Tjong Kim Sang, E.F.: Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. Proceedings of CoNLL-2002; 155-158.

[WI95] Winkler, W.E.: Matching and record-linkage. In Business Survey Methods, John Wiley & Sons, 1995; 355-384.

[ZO96] Zobel, J.; Dart, P.: Phonetic string matching: Lessons from Information Retrieval. Proceed. 19th Annual Intl. ACM SIGIR Conf. on R&D in IR. 1996; 166-173.