Available online at www.sciencedirect.com
ScienceDirect
Journal of Informetrics 1 (2007) 35–46
Journal of INFORMETRICS
An International Journal
www.elsevier.com/locate/joi

Measuring quality of similarity functions in approximate data matching

Roberto da Silva*, Raquel Stasiu[1], Viviane Moreira Orengo, Carlos A. Heuser

UFRGS, Instituto de Informática, Porto Alegre, Brazil

Received 3 July 2006; received in revised form 23 August 2006; accepted 5 September 2006

Abstract

This paper presents a method for assessing the quality of similarity functions. The scenario taken into account is that of approximate data matching, in which it is necessary to determine whether two data instances represent the same real world object. Our method is based on the semi-automatic estimation of optimal threshold values. We propose two methods for performing such estimation. The first method is an algorithm based on a reward function, and the second is a statistical method. Experiments were carried out to validate the techniques proposed. The results show that both methods for threshold estimation produce similar results. The output of such methods was used to design a grading function for similarity functions. This grading function, called *discernability*, was used to compare a number of similarity functions applied to an experimental data set.

Keywords: Approximate data matching; Similarity functions; Retrieval evaluation

1. Introduction

The process of approximate data matching aims at defining whether two data instances (strings, tuples, trees, ...) represent the same real world object. This process appears in several data management applications such as *approximate querying* and *data integration*. In approximate querying, the problem is to find database instances that represent the same data instance given as a query. In data integration, the aim is to assess whether two data instances originating from different sources represent the same real world object. Approximate data matching usually relies on the use of a *similarity function*. A similarity function $f(v_1, v_2) \mapsto s$ assigns a score s to a pair of data values v_1 and v_2. These values are considered to be representing the same real world object if s is greater then a given *threshold* t. There is a wide range of similarity functions, from very simple string matching functions, like Levenshtein's edit distance (Hall & Dowling, 1980; Levenshtein, 1966; Navarro, Baeza-Yates, Sutinen, & Tarhio, 2001), to functions specific to XML trees (Dorneles, Heuser, Lima, da Silva, & de Moura, 2004). Generally speaking, similarity functions are imperfect and the quality of their results will depend on the specific data set being matched.

The use of similarity functions in approximate data matching poses two problems. The first is to determine the threshold value that should be used. The difficulty in this case arises from the fact that the distribution of score values

* Corresponding author. Tel.: +55 51 33167772; fax: +55 51 3316 7308.
E-mail addresses: rdasilva@inf.ufrgs.br (R. da Silva), rkstasiu@inf.ufrgs.br (R. Stasiu), vmorengo@inf.ufrgs.br (V.M. Orengo), heuser@inf.ufrgs.br (C.A. Heuser).
[1] On leave from PUC-PR and UTFPR.

1751-1577/$ – see front matter
doi:10.1016/j.joi.2006.09.001

obtained by one similarity function may be completely different from the distribution obtained by another. It may even vary when the same similarity function is applied to different data sets.

The second problem is how to measure if a similarity function is more adequate for a specific data set then another. Existing approaches for the evaluation of similarity functions (Bilenko, Mooney, Cohen, Ravikumar, & Fienberg, 2003; Cohen, 2003) are based on the recall/precision curve, a classical Information Retrieval (IR) quality measure (Salton, 1989). Recall/precision curves are useful to express the ability of a similarity function in ranking the results of matches. However, they are not suitable for expressing how efficient similarity functions are in telling apart relevant from irrelevant matches.

In this paper, we propose a quality measure specifically designed for similarity functions in the context of data matching. As a byproduct, our approach also produces a threshold value that may be interpreted as the "best" one for a given similarity function, when considering a specific data set. Here, "best" means a threshold value that minimizes false positives and false negatives with respect to an answer set.

The remainder of this paper is organized as follows: Section 2 proposes two methods for threshold definition; Section 3 presents experiments that evaluate the proposed methods; Section 4 proposes a function called *discernability* to assess the quality of similarity functions and applies it to compare several similarity functions; Section 5 presents a summary and the conclusions.

2. Process of threshold definition

For most applications, the process of threshold definition is left to the user who must choose an arbitrary value to be applied to one or more queries. If the threshold chosen is too high, there is a risk of not retrieving any results. On the other hand, if the chosen threshold is too low, many irrelevant items will be retrieved. This problem is aggravated by the fact, mentioned in the introduction, that the distribution of score values may vary significantly from one similarity function to another. As a result, the definition of a threshold is generally a trial and error process, in which the user has to test a number of different values until the result is satisfactory.

In this section, we propose two semi-automatic methods for the calculation of threshold values for a given similarity function. The output of these methods is an interval of threshold values $[t_{\text{best}}^{\min}, t_{\text{best}}^{\max}]$ that provides optimum results. By optimum we mean a threshold value that maximizes the number of cases in which $s_{\text{irrel}} \leq t_{\text{best}} \leq s_{\text{rel}}$, where s_{rel} is the lowest score for a relevant item, s_{irrel} is the highest score for an irrelevant item. In what follows, we explain how to calculate these scores. The process of threshold definition should be guided by two premises: (i) minimize false positives and (ii) minimize false negatives.

Both methods rely on a sampling process that takes values from a pre-existing *collection* V of data values (v) to be compared by a similarity function. These sample values are then used as queries against V in order to collect knowledge about how the score values are distributed.

More specifically, the sampling process is as follows: A *sample* $Q \subseteq V$ is taken from the collection. Each element $q \in Q$ is used as a query object against the collection V. The similarity between the query and each element of the database is calculated using a given similarity function:

$$L : (Q \subseteq V) \times V \to \mathbb{R}^+.$$

So, for each $q \in Q$ we define the set:

$$R_q = \{s \in \mathbb{R}^+ / s = L(q, v),\ v \in V\}.$$

Here, R_q induces an order $\prec$ on V, defined by relation:

$$v, w \in V, \quad v \prec w \Leftrightarrow L(q, v) < L(q, w),$$

then the values in V are ranked in decreasing order of similarity to the query value q.

Next, a human expert labels each element of the ranking as `relevant (rel)`, if the data value is considered to represent the same real world object as the query q or as `irrelevant (irrel)` otherwise. Defining

$$v_q(\text{rel}) = \min\{v / v \text{ is relevant}\}, \qquad v_q(\text{irrel}) = \max\{v / v \text{ is irrelevant}\},$$

if $n = |Q|$, $q \in Q$ we note k the index of q such that $k \in [1, n]$.

Table 1
Example of similarity ranking

Score	Data item	Relevance
1.0000	Journal of Informetrics	Relevant
0.8636	Jrnl of Infometrics	Relevant
0.7391	J. of Informetrics	Relevant
0.1304	Informetrics Journal	Relevant
0.1304	JOI	Relevant
0.1250	Decision Support Systems	Irrelevant
0.0869	TODS	Irrelevant
0.0869	SIGMOD	Irrelevant
0.0434	TKDE	Irrelevant

This labelling enables us to identify two important points in the ranking: $s^L_{\text{rel}}(k) = L(q, v_q(\text{rel}))$, which is the lowest score corresponding to a relevant item and $s^L_{\text{irrel}}(k) = L(q, v_q(\text{irrel}))$, which is the highest score attained by an irrelevant item. Those values are used by both methods for threshold definition proposed in this paper. Notice that for some queries $s^L_{\text{irrel}}(k)$ could be greater than $s^L_{\text{rel}}(k)$. Such a situation indicates that the similarity function has failed to separate relevant from irrelevant items.

Example. Consider a database containing titles of computing science journals. The object "Journal of Informetrics" is represented in five different forms, namely: "Journal of Informetrics", "J. of Informetrics", "JOI", "Informetrics Journal", "Jrnl of Infometrics". Supposing that the database contains nine data items, the ranking generated by the edit distance function is shown in Table 1. According to this ranking, the lowest score of a relevant item is $s_{\text{rel}} = 0.1304$ and the highest score of an irrelevant item is $s_{\text{irrel}} = 0.1250$.

2.1. Reward function algorithm

A function to measure how good a threshold value is in separating relevant from irrelevant items can be defined by the simple formula below:

$$f^L(n, t) = \sum_{k=1}^{n} d(s^L_{\text{rel}}(k), s^L_{\text{irrel}}(k)) \tag{1}$$

where L is the similarity function used; n the number of queries (sample size); t is the threshold being analyzed; $d(\cdot, \cdot)$ measures how adequate $s^L_{\text{rel}}(k)$ and $s^L_{\text{irrel}}(k)$ are with the threshold t, such that:

$$d(s^L_{\text{rel}}(k), s^L_{\text{irrel}}(k)) = R^t_{\text{rel}}(k) + R^t_{\text{irrel}}(k) \tag{2}$$

with

$$R^t_{\text{rel}}(k) = \begin{cases} 1 & \text{if } s^L_{\text{rel}}(k) > t \\ -1 & \text{else } s^L_{\text{rel}}(k) \le t \end{cases} \quad \text{and} \quad R^t_{\text{irrel}}(k) = \begin{cases} -1 & \text{if } s^L_{\text{irrel}}(k) \ge t \\ 1 & \text{else } s^L_{\text{irrel}}(k) < t \end{cases} \tag{3}$$

According to these equations, the optimal threshold t_{best} (or more precisely the interval for the optimal threshold), which reaches the maximum value on the function $f^L(n, t)$, can be defined as

$$f^L_{\max} = \max_{t \in [t_{\min}, t_{\max}]} \{f^L(n, t)\}. \tag{4}$$

where $t_{\min}$ and $t_{\max}$ represent the limits of the threshold interval to be tested.

Algorithm 1 shows a description of BestThresh, which determines t_{best}. The inputs for this algorithm are: (i) the number of queries (n); (ii) the limits of the threshold interval to be tested ($t_{\min}$ and $t_{\max}$); (iii) the lowest similarity score achieved by a relevant item for the query k, denoted by $s^L_{\text{rel}}(k)$; (iv) the highest score achieved by an irrelevant item for the same query k, denoted by $s^L_{\text{irrel}}(k)$; (v) the numerical precision (h) on which the algorithm should operate. The algorithm produces two outputs: the interval $[t^{\min}_{\text{best}}, t^{\max}_{\text{best}}]$ in which the optimal threshold

(t_{best}) lies; and its associated f_{max}, which is the number of points achieved by that threshold interval. The reason for the output of the algorithm being an interval and not a single value is that a number of threshold values, in sequential order, can achieve f_{max}. The lowest and the highest values are then used as limits of the interval. Ways in which f_{max} could be used in the evaluation of the quality of the similarity function will be discussed in Section 4.

The limits t_{min} and t_{max} are, respectively, the smallest and the largest similarity scores from the ranking generated by the similarity function. The numerical precision, denoted by h, is calculated by the formula $h = (t_{max} - t_{min})/n_{div}$, where n_{div} is the number of divisions we want to make on the interval $[t_{min}, t_{max}]$. This way, each threshold t to be tested by the algorithm is obtained by $t_i = t_{min} + ih$, where $i = 0, \ldots, n_{div}$.

The algorithm works as follows: each threshold t between t_{min} and t_{max} is tested for each query. The test consists in comparing t with s_{rel} and s_{irrel}. The number of points achieved by each threshold t is computed according to Eqs. (3) and (4). The highest number of points achieved by a threshold (f_{max}), which is initialized at the beginning of the algorithm with the smallest value possible, is then found. Once f_{max} is established, the algorithm finds the interval in which all threshold values achieve f_{max}.

Algorithm 1. BestThresh

Input: $n, t_{min}, t_{max}, s^L_{rel}(k), s^L_{irrel}(k), k = 1 \ldots n, h$
Output: $t^{min}_{best}, t^{max}_{best}, f_{max}$
$f_{max} = -2n$;
$n_{div} = (t_{max} - t_{min})/h$
for (a) $i = 0, \ldots, n_{div}$ **do**
  $t = t_{min} + ih$;
  $f(t) = 0$;
  for (b) $k = 1, \ldots, n$ **do**
    $d = 0$;
    if ($s^L_{rel}(k) > t$) **then**
      $d = d + 1$;
    else
      $d = d - 1$;
    end if
    if ($s^L_{irrel}(k) < t$) **then**
      $d = d + 1$;
    else
      $d = d - 1$;
    end if
    $f(t) = f(t) + d$;
  end for (b)
  if ($f(t) \geq f_{max}$) **then**
    $f_{max} = f(t)$;
  end if
end for (a)
$t = t_{min}$
while ($f(t) \neq f_{max}$) **do**
  $t = t + h$
end while
$t^{min}_{best} = t$
$t = t_{max}$
while ($f(t) \neq f_{max}$) **do**
  $t = t - h$
end while
$t^{max}_{best} = t$
if $f_{max} < 0$ **then**
  aux $= t^{max}_{best}$
  $t^{max}_{best} = t^{min}_{best}$
  $t^{min}_{best} =$ aux
end if
Write "the best threshold is in the interval" $[t^{min}_{best}, t^{max}_{best}]$

2.2. Bivariate normal distribution

In this section, we explore the approach of a bivariate normal distribution (Spiegel, 1992; Weisstein, 2004) to find t_{best}. This method is based on statistics and tries to maximize the probability of finding a threshold that minimizes false positives and false negatives.

Let us consider the probability density function (PDF) for s^L_{irrel} and s^L_{rel}, denoted by $P(s^L_{\text{irrel}})$ and $P(s^L_{\text{rel}})$, respectively. Considering a sample of size n, the experimental mean values are computed as $\langle s^L_{\text{irrel}}\rangle = (1/n)\sum_{k=1}^{n} s^L_{\text{irrel}}(k)$ and $\langle s^L_{\text{rel}}\rangle = (1/n)\sum_{k=1}^{n} s^L_{\text{rel}}(k)$, and the respective standard deviation $\sigma(s^L_{\text{irrel}}) = \sqrt{[1/(n-1)]\sum_{k=1}^{n}[s^L_{\text{irrel}}(k) - \langle s^L_{\text{irrel}}\rangle]^2}$, $\sigma(s^L_{\text{rel}}) = \sqrt{[1/(n-1)]\sum_{k=1}^{n}[s^L_{\text{rel}}(k) - \langle s^L_{\text{rel}}\rangle]^2}$.

Given the means and the standard deviations, we can calculate the distributions for $P(s_{\text{rel}})$ and $P(s_{\text{irrel}})$, which would approximately be:

$$P(s^L_{\text{irrel}}) = \frac{1}{\sqrt{2\pi\sigma^2(s^L_{\text{irrel}})}} \exp\left[-\frac{1}{2}\left(\frac{s^L_{\text{irrel}} - \langle s^L_{\text{irrel}}\rangle}{\sigma(s^L_{\text{irrel}})}\right)^2\right],$$

$$P(s^L_{\text{rel}}) = \frac{1}{\sqrt{2\pi\sigma^2(s^L_{\text{rel}})}} \exp\left[-\frac{1}{2}\left(\frac{s^L_{\text{rel}} - \langle s^L_{\text{rel}}\rangle}{\sigma(s^L_{\text{rel}})}\right)^2\right] \tag{5}$$

Since there is a correlation between $s^L_{\text{irrel}}(k)$ and $s^L_{\text{rel}}(k)$, the joint distribution $P(s^L_{\text{irrel}}, s^L_{\text{rel}})$ is not necessarily the product $P(s^L_{\text{irrel}}) \cdot P(s^L_{\text{rel}})$. Therefore, in order to calculate the joint PDF, we need to take the correlation coefficient ρ into consideration. The formula for the joint PDF is given below:

$$P(s^L_{\text{irrel}}, s^L_{\text{rel}}) = \frac{1}{2\pi\sigma(s^L_{\text{rel}})\sigma(s^L_{\text{irrel}})\sqrt{1-\rho^2}} \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{s^L_{\text{irrel}} - \langle s^L_{\text{irrel}}\rangle}{\sigma(s^L_{\text{irrel}})}\right)^2 + \left(\frac{s^L_{\text{rel}} - \langle s^L_{\text{rel}}\rangle}{\sigma(s^L_{\text{rel}})}\right)^2 - 2\rho\left(\frac{s^L_{\text{irrel}} - \langle s^L_{\text{irrel}}\rangle}{\sigma(s^L_{\text{irrel}})}\right)\left(\frac{s^L_{\text{rel}} - \langle s^L_{\text{rel}}\rangle}{\sigma(s^L_{\text{rel}})}\right)\right]\right\}, \tag{6}$$

where ρ is the correlation coefficient defined by the formula:

$$\rho = \frac{\langle s^L_{\text{rel}} s^L_{\text{irrel}}\rangle - \langle s^L_{\text{rel}}\rangle\langle s^L_{\text{irrel}}\rangle}{\sigma(s^L_{\text{rel}})\sigma(s^L_{\text{irrel}})}, \tag{7}$$

that assumes values in the interval $[-1, 1]$, where $|\rho| \sim 1$ denotes correlated data and $|\rho| \sim 0$ denotes that in our data, s^L_{rel} and s^L_{irrel} are independent random variables.

For determining t_{best}, it is sufficient to find the value of t that yields the maximum:

$$F(t) = P(s^L_{\text{irrel}}\langle t, s^L_{\text{rel}}\rangle t) = \frac{1}{2\pi\sigma(s^L_{\text{rel}})\sigma(s^L_{\text{irrel}})\sqrt{1-\rho^2}} \int_{-\infty}^{t}\int_{t}^{\infty} \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\left(\frac{s^L_{\text{irrel}} - \langle s^L_{\text{irrel}}\rangle}{\sigma(s^L_{\text{irrel}})}\right)^2 + \left(\frac{s^L_{\text{rel}} - \langle s^L_{\text{rel}}\rangle}{\sigma(s^L_{\text{rel}})}\right)^2 - 2\rho\left(\frac{s^L_{\text{irrel}} - \langle s^L_{\text{irrel}}\rangle}{\sigma(s^L_{\text{irrel}})}\right)\left(\frac{s^L_{\text{rel}} - \langle s^L_{\text{rel}}\rangle}{\sigma(s^L_{\text{rel}})}\right)\right]\right\} \mathrm{d}s^L_{\text{irrel}}\,\mathrm{d}s^L_{\text{rel}} \tag{8}$$

That is, we are trying to find the value of t that maximizes the probability of t being simultaneously greater than s^L_{irrel} and less than s^L_{rel}. A simple algorithm was implemented to compute $F(t)$, in order to discover the value of t_{best} for each similarity function used in the experiments.

3. Experiments

In this section, we describe the experiments we carried out in order to evaluate the two threshold definition methods proposed in Section 2. We collected titles for 18 scientific papers and manually edited them (i.e. adding, replacing, removing and/or swapping characters or words) to simulate possible typing errors. In total 150 paper titles were

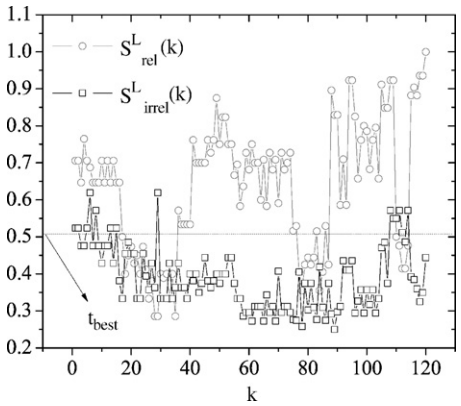

Fig. 1. Lowest relevant score and highest irrelevant score as function of the kth query for function L (edit distance).

generated. A sample of 120 items was picked and used as queries against the database. The similarity between the each query and the documents was calculated using the Edit distance function (Hall & Dowling, 1980; Levenshtein, 1966; Navarro et al., 2001). This function calculates the minimum number of character insertions, deletions and replacements necessary to make two strings equal.

As mentioned in Section 2, a human expert labelled all returned items in the ranked list as relevant or irrelevant. Based on this labelling, the values for $s^L_{\text{rel}}(k)$ and $s^L_{\text{irrel}}(k)$ were obtained. Fig. 1 shows a plot of $s^L_{\text{rel}}(k)$ and $s^L_{\text{irrel}}(k)$ as a function of the kth query, for a sample of 120 queries (n).

3.1. Experiments using the BestThresh algorithm

Considering a number from $k = 1$ to n queries and a precision of $h = (t_{\max} - t_{\min})/n_{\text{div}} = 0.001$, where $t_{\max} = 1$, $t_{\min} = 0$ and $n_{\text{div}} = 1000$, we run the BestThresh algorithm evaluating each threshold calculated by the formula $t_i = t_{\min} + ih$, where $i = 1, \ldots, n_{\text{div}}$. The results produced by the algorithm indicate that t_{best} lies in the interval $I = [0.524, 0.529]$. All threshold values within this interval have achieved $f_{\max} = f^{\text{edit}}(n, t) = 154$. Thus, whichever value belonging to I would be a suitable threshold for performing a search by chance using this particular similarity function. Fig. 2 shows a plot of $f^{\text{edit}}(n, t)$ as a function t.

Notice that the values of $f^{\text{edit}}(n, t)$ are distributed symmetrically as function of t. The continuous curve in Fig. 2 is a normal fit for our data, which gives us an exact notion of how our values are distributed.

A "robustness" test can be performed to assess how t_{best} behaves as the sample size (n) grows. Intuitively, t_{best} should converge to a constant value t^{∞}_{best} when extrapolating $k \to n$. Fig. 3 shows that t_{best} stabilizes as the number of queries approaches 120.

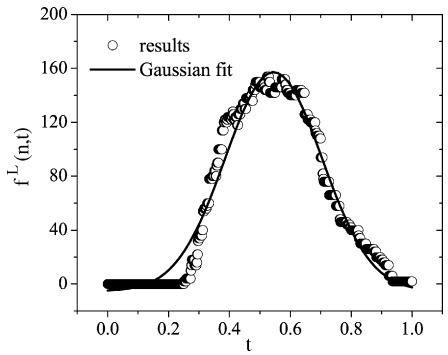

Fig. 2. Plot of $f^{\text{edit}}(n, t)$ as function of t for the edit distance function.

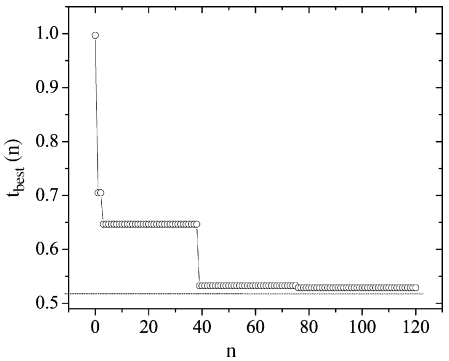

Fig. 3. Evolution of t_{best} as function of sample size. The plot clearly shows that t_{best} converges to the values in the interval $I = [0.524, 0.529]$ as $n \to \infty$.

3.2. Results using bivariate normal distribution

We calculated the parameters $(\langle s^L_{irrel}\rangle, \sigma^2(s^L_{irrel}))$ and $(\langle s^L_{rel}\rangle, \sigma^2(s^L_{rel}))$ for the sample queries using the same similarity function applied in the previous subsection (edit distance). Histograms for the values for s^L_{irrel} and s^L_{rel} were also computed. Fig. 4 shows how s^L_{irrel} and s^L_{rel} are distributed around the mean values $\langle s^L_{irrel}\rangle$ and $\langle s^L_{rel}\rangle$. The continuous curves in these plots denote the normal fits. Calculating the correlation $\rho = 0.022$, we obtained the bivariate normal, shown in Fig. 5.

The numerical software Maple was used to calculate $F(t)$ specified by Eq. (8), spanning t in the interval $[0, 1]$ to find the value of t that yields the maximum $F(t)$. We present our results on Fig. 5 as a plot of $F(t)$ as a function of t. Notice that the probability $F(t)$ is approximately a normal PDF once that the continuous curve is a normal PDF fit. The figure shows that the most likely value for t_{best} is 0.515. This value was obtained by our statistical method

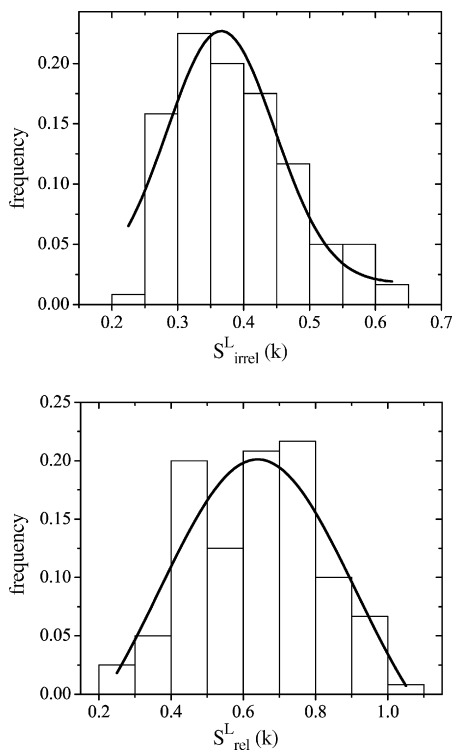

Fig. 4. Histograms for the values of s^L_{irrel} and s^L_{rel}. The continuous curves are the gaussian fits for these histograms. A correlation coefficient between the two distributions can be determined.

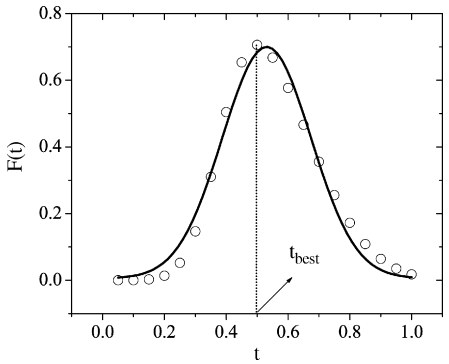

Fig. 5. Plot $F(t) \times t$. The most likely value for t_{best} is the value of t corresponding to the highest value of $F(t)$.

considering a precision of $h = 0.001$. Recall that the value for t_{best} calculated by the BestThresh algorithm was in the interval $I = [0.524, 0.529]$. This shows that both methods are in agreement.

4. Evaluating similarity functions

The aim of this section is to use our two methods for threshold definition to evaluate the quality of different similarity functions. One similarity function can be considered better than another if it provides better separation of relevant and irrelevant data items returned in response to a query. According to our approach, a similarity function that has a higher $f_{\max}$ is considered better than another function that has a smaller $f_{\max}$. Also, the size of the range of the interval for t_{best} is another indicator of the quality of the function. Given that a good similarity function should place relevant and irrelevant items far apart in the ranking, the larger the interval, the better. Section 4.1 proposes a function that evaluates the quality of a similarity function for a specific data set. Section 4.2 applies the proposed discernability to assess a number of similarity functions.

4.1. The discernability function

Below we define a method for assessing the quality of a similarity function. We named the proposed function *discernability* as it refers to the ability of the similarity function in discerning relevant from irrelevant items. Discernability takes two aspects into consideration: (i) how well the similarity function separates relevant from irrelevant items; (ii) how far apart in the ranking the similarity function places relevant and irrelevant items. The first aspect is given by the maximum number of points ($f_{\max}$) calculated by the BestThresh algorithm. The second aspect can be calculated by taking the difference between $t_{\text{best}}^{\max}$ and $t_{\text{best}}^{\min}$. Discernability also defines two coefficients c_1 and c_2 which allow the user to express the importance given to each of the two aspects considered. For the experiments described in this paper we gave the same importance to c_1 and c_2 using $c_1 = c_2 = 1$. The values produced by the discernability will be in the interval $[-1, 1]$.

$$discernability^{L}(t_{\text{best}}^{\min}, t_{\text{best}}^{\max}, f_{\max}) = \frac{c_1}{c_1 + c_2}(t_{\text{best}}^{\max} - t_{\text{best}}^{\min}) + \frac{c_2}{c_1 + c_2} \cdot \frac{f_{\max}}{2n} \tag{9}$$

In order to assess whether the threshold values calculated by our two methods are plausible, we define two measures for computing the theoretical confidence interval, considering the distribution of threshold values. In this case the average value of t is given by

$$\langle t \rangle = \frac{\sum_{i=1}^{n} t_i F(t_i)}{\sum_{i=1}^{n} F(t_i)} \tag{10}$$

and the respective uncertainty associated

$$\sigma_t = \sqrt{\frac{\sum_{i=1}^{n} t_i^2 F(t_i)}{\sum_{i=1}^{n} F(t_i)} - \langle t \rangle^2} \tag{11}$$

4.2. Experiments

The same data used in Section 3 was used to compare the performance of eight similarity functions. Below we list each function together with a brief description.

- **Edit distance** (Hall & Dowling, 1980; Levenshtein, 1966; Navarro et al., 2001). As mentioned in the previous section, this function computes the minimum number of changes (insertions,deletions and replacements) that are necessary to make two strings equal.
- **Acronyms** (Dorneles et al., 2004). This function is useful for matching acronyms to their unabbreviated form, e.g. matching "JOI" to "Journal of Informetrics".
- **Guth** (Guth, 1976). This function is designed for matching proper nouns.
- **Jaccard** (Jaccard, 1912). This simple function states that the similarity between s_1 and s_2 is given by $(s_1 \cap s_2) \div (s_1 \cup s_2)$.
- **Jaro** (Jaro, 1989). This is a function based on the number and order of common characters between two strings.
- **JaroWinkle**r (Winkler, 1999). This is a variant of the Jaro function that emphasizes matches in the first few characters.
- **N-gram** (Navarro et al., 2001). The similarity score is calculated based on the number of characters that are in the same position in each gram. For the experiments described in this section, we used $n = 3$.
- **TF-IDF** (Salton & McGill, 1983). The acronym stands for term-frequency inverse document frequency. This is widely used in IR as a weighting scheme in order to give more importance to less frequent words. For string matching, TF is the frequency of the term in the string and IDF can be computed using the entire collection of strings to be matched.

In addition to the real similarity functions above, we tested three artificial ones:

- **Optimal**. The perfect similarity function should correctly separate relevant and irrelevant items and place them as far as possible in the ranking, i.e. for all queries $s_{\text{rel}} = 1$ and $s_{\text{irrel}} = 0$.
- **NoneRetrieved**. Function that calculates a similarity score of zero between the query and all data items, no matter whether they are relevant or not. In this case for all queries $s_{\text{rel}} = s_{\text{irrel}} = 0$.
- **WorstPossible**. The worst function places all non-relevant items higher than the relevant ones in the ranking, i.e. for all queries $s_{\text{rel}} = 0$ and $s_{\text{irrel}} = 1$.

A precision of $h = 0.001$ was used for the computation of the interval $[t_{\text{best}}^{\min}, t_{\text{best}}^{\max}]$ by the BestTresh algorithm. The limits of this interval were used to calculate the *discernability* for the similarity function. t_{best} was calculated by the bivariate normal distribution.

The results are shown in Table 2. The second column of the table presents the values for $f_{\max}$, which represent the number of points achieved by t_{best} for a given function. The third column displays the results for *discernability*. The fourth column shows the interval for t_{best} calculated by the BestThresh algorithm. The fifth column contains the most

Table 2
Comparison among different similarity functions

Function	$f_{\max}$	Discernability	$[t_{\text{best}}^{\min}, t_{\text{best}}^{\max}]$	t_{best}	Confidence interval
Jaro–Winkler	184	0.4048	[0.768, 0.811]	0.791	[0.716, 0.887]
Jaro	178	0.3713	[0.755, 0.756]	0.753	[0.703, 0.888]
Acronyms	158	0.3616	[0.601, 0.666]	0.592	[0.455, 0.781]
Edit distance	154	0.3233	[0.524, 0.529]	0.515	[0.404, 0.697]
N-gram	134	0.2851	[0.576, 0.588]	0.553	[0.440, 0.723]
Guth	38	0.0821	[0.905, 0.911]	0.801	[0.686, 0.896]
Jaccard	16	0.0468	[0.401, 0.428]	0.301	[0.169, 0.428]
TFIDF	16	0.0438	[0.578, 0.599]	0.442	[0.273, 0.614]
Optimal*	240	0.9999	[0.001, 0.999]	–	–
NoneRetrieved*	0	0.0000	[0.000, 0.000]	–	–
WorstPossible*	−240	−0.9999	[0.999, 0.001]	–	–

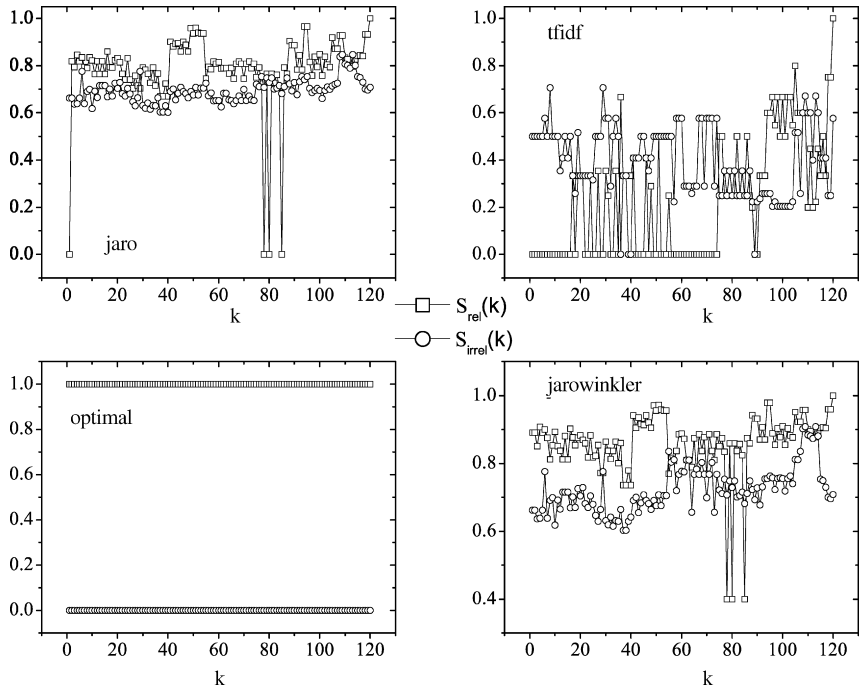

Fig. 6. Distribution of S_{rel} and S_{irrel} for different similarity functions.

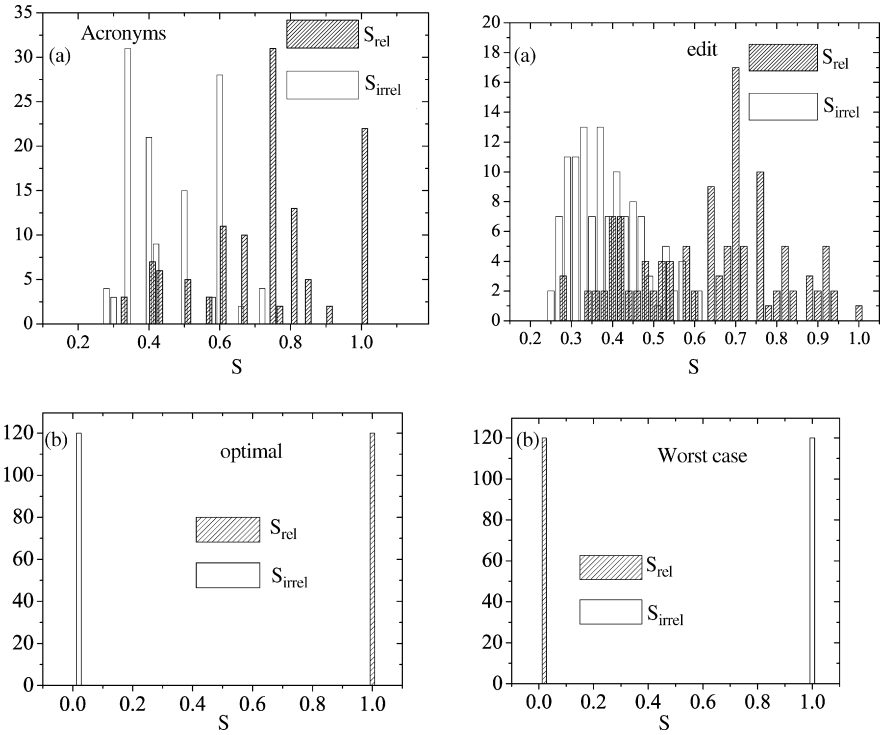

Fig. 7. The x axis represents the threshold values and the y axis represents the number of occurrences, i.e. how many queries achieved that threshold value for s_{rel} and s_{irrel}. The plots tagged (a) show examples of histograms for functions that are statistically treatable and the plots tagged (b) illustrate histograms for functions that are not statistically treatable.

likely value for t_{best} computed by our statistical method for threshold definition. In the last column of the table, we show the confidence interval built with the distribution $F(t)$ by Eqs. (10) and (11).

Table 2 shows that the absolute difference between the results of our two proposed methods for threshold estimation is at most 0.136, showing that the two approaches are in agreement. It is worth pointing out that the better the similarity function, the more in agreement the two methods are. Furthermore, in all cases the values calculated by both methods are within the theoretical confidence interval.

Table 2 also shows the results for discernability. According to them, the best real function for the data set analyzed was Jaro–Winkler and the worst was TFIDF. This can be confirmed by observing the plots in Fig. 6, which show the distribution of s_{rel} and s_{irrel}. Indeed, the best separation between relevant and irrelevant data items was achieved by Jaro–Winkler, whilst with TFIDF these items are often shuffled and/or too close together in the ranking. The plots also show the behavior of the (artificial) Optimal function, which would achieve the highest marks according to the discernability function. The behavior of Jaro, which achieved the second best result, is also plotted in Fig. 6. It is worth pointing out that this ranking is for the data set used in the experiment. For a different data set, the ranking would most probably differ.

The similarity functions in Table 2 that have the symbol * are functions for which the statistical approach is not applicable due to the nature of the data, i.e., there is no variability in the measures of similarity using this function. By no variability we mean that the values for s_{rel} and/or the values for s_{irrel} are constant for most queries. In other words, the standard deviations for s_{rel} and $s_{irrel}(\sigma(s^L_{rel})$ and $\sigma(s^L_{irrel}))$ are close to zero. Nevertheless, it is still possible to find an optimal threshold using the BestThresh algorithm.

In Fig. 7 we present plots of dispersion for the similarity values using two types of functions. Type (a) represent functions that are statistically treatable (or in which there is a reasonable variability in the data); and type (b) represent functions which are not statistically treatable.

5. Summary and conclusions

The contributions of this paper are two-fold. Our goal was to propose a method for measuring the quality of similarity functions in separating relevant from irrelevant data items returned in response to a query. In order to achieve this goal, we made a second contribution which was the development of techniques for the estimation of optimal threshold values. Such techniques can be applied not only in the evaluation of similarity functions but also in standard IR experiments to assess the quality of different ranking algorithms.

Several experiments were carried out in order to evaluate our proposed approaches. Initially, we performed experiments to test the threshold definition methods. The results show that both techniques produce similar values, validating one another.

In this paper, we used human intervention to identify relevant and irrelevant data items. However, it is worth pointing out that this sampling process could be automated through the use of clustering algorithms, as done in our previous work (Stasiu, Heuser, & da Silva, 2005). In this case, all the elements of a given cluster are considered as representing the same real world object. Thus, the relevant results for a query are the elements from the same cluster as the query. The sampling (or clustering) phase can be seen as a type of training. After this process, new queries should produce better results as a consequence of the use of a more suitable threshold.

We used the output produced by our threshold definition methods to design a "grade", which we called discernability, to measure the quality of similarity functions. The discernability takes into consideration the separation and the distance between relevant and irrelevant items. Those two aspects may be weighted differently according to their importance in the data set being analyzed. Finally, we performed experiments to assess the quality of eight similarity functions according to the discernability function. The results show that, for the data set considered, the best function was Jaro-Winkler and the worst was TFIDF.

Acknowledgments

We would like to thank the anonymous referees for the helpful suggestions.

This work was partially financed by the following projects: SisTol-CNPq, CAPES-GRICES (finished), PROBRAL-CAPES, Gerindo (CNPq/CTInfo55.2087/2002-5), XMLBroker (CNPq/Universal473310/2004-0), Rec-Semântica (FAPERGSCNPq/PRONEX-2004), Digitex (CNPq/CT-Info550845/2005-4), a PhD Scholarship from CAPES, and CAPES-PRODOC.

References

Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., & Fienberg, S. (2003). Adaptive name matching in information integration. *IEEE Intelligent Systems*, *18* (5), 16–23.

Cohen, W., Ravikumar, P., & Fienberg, S. (2003). A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IIWeb* (pp. 73–78).

Dorneles, C. F., Heuser, C. A., Lima, A. E. N., da Silva, A. S., & de Moura, E. S. (2004). Measuring similarity between collection of values. In *WIDM '04: Proceedings of the sixth annual ACM international workshop on Web information and data management* (pp. 56–63). New York, NY, USA: ACM Press.

Guth, G. J. (1976). Surname spellings and computerized record linkage. *Historical Methods Newsletter*, *10* (1), 10–19.

Hall, P. A. V., & Dowling, G. F. (1980). Approximate string matching. *ACM Computing Surveys*, *12* (4), 381–402.

Jaccard, P. (1912). The distribution of flora in the alpine zone. *New Phytologist*, *11* (2), 37–50.

Jaro, M. (1989). Advances in record linking methodology as applied to the 1985 census of Tampa Florida. *Journal of the American Statistical Society*, *64*, 1183–1210.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, *10* (8), 707–710.

Navarro, G., Baeza-Yates, R., Sutinen, E., & Tarhio, J. (2001). Indexing methods for approximate string matching. *IEEE Data Engineering Bulletin*, *24* (4), 19–27.

Salton, G. (1989). *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. New York, NY, USA: McGraw-Hill.

Spiegel, M. R. (1992). *Theory and problems of probability and statistics*. McGraw-Hill.

Stasiu, R.K., Heuser, C.A., & da Silva, R. (2005). Estimating recall and precision for vague queries in databases. In *CAISE05: Proceedings of the 17th conference on advanced information systems engineering* (pp. 187–200). Springer Verlag, Porto, Portugal, June 13–17, 20, Lecture Notes in Computer Science.

Weisstein, E. W. (2004). *Bivariate normal distribution. From MathWorld—A Wolfram Web Resource*. Last modification: URL http://mathworld.wolfram.com/BivariateNormalDistribution.html.

Winkler, W. (1999). The state of record linkage and current research problems. In *Statistics of Income Division, Internal Revenue Service Publication R99/04*. URL www.census.gov/srd/www/byname.html.