SimEval - A Tool for Evaluating the Quality of Similarity Functions

Carlos A. Heuser **Francisco N. A. Krieser** **Viviane Moreira Orengo**

UFRGS - Instituto de Informtica
Caixa Postal 15.064 - 91501-970 - Porto Alegre - RS - Brazil
Email: [heuser, fkrieser, viviane]@inf.ufrgs.br

Abstract

Approximate data matching applications typically use similarity functions to quantify the degree of likeness between two data instances. There are several similarity functions available, thus, it is often necessary to evaluate a number of them aiming at choosing the function that is more adequate to a specific application. This paper presents a tool that uses average precision and discernability to evaluate the quality of similarity functions over a data set.

Keywords: approximate data matching, similarity functions

1 Introduction

Approximate or similarity matching deals with the problem of assessing if two different data instances represent the same real world entity. Approximate matching plays a central role in several advanced data management applications like similarity join (Cohen 2000), entity resolution or record deduplication (Benjelloun et al. 2006) and schema matching.

Approximate matching usually relies on *similarity functions* to evaluate whether two data instances (strings, tuples, etc) represent the same real world entity. A similarity function $f(v_1, v_2) \rightarrow s$ takes two data instances v_1 and v_2 as inputs and calculates a score value s between 0 and 1. If the value for s surpasses a given threshold t, the values v_1 and v_2 are considered to be representations of the same real world entity. There is a large variety of similarity functions, from simple string metrics (such as Levenshtein's edit distance (Levenshtein 1966)) to metrics specific to XML trees (Dorneles et al. 2004). Due to this variety, designers often meet the task of choosing the most appropriate function for a given application.

One measure often employed for evaluating and comparing similarity functions is the classical *mean average precision* (*MAP*). MAP measures the ability that a similarity function has in returning higher similarity values for relevant matches and returning lower similarity values for irrelevant matches. Using the terminology applied in the information retrieval area, MAP measures the ability that a similarity function has in moving relevant data instances to the top of the ranking of results of a query. This means that MAP is an adequate measure for evaluating similarity functions in applications that employ *top-k* queries. Top-k queries return the k data instances that are most similar to the query object. This is the typical query in many information retrieval applications.

However, the queries employed in the data matching applications are usually *range* queries. A range query returns all data instances that when compared with a query data instance return a similarity score that is greater than a specified threshold value. This means that when evaluating similarity functions for range query applications one must take in account not only the ability that the similarity function has in returning relevant values at the top of a ranking, but also the ability that the function has in returning score values that are greater than the threshold for relevant matches and returning score values that are lower than the threshold for matches items. *Discernability* (da Silva et al. 2007) is a measure specifically designed for evaluating the quality of similarity functions for range queries. An important difference between these *discernability* and MAP is that discernability takes the scores assigned to relevant and irrelevant data items into consideration and not just their ranks as MAP does.

This paper presents *SimEval*, a tool that evaluates the quality of similarity functions for a given data set. In order to perform such evaluation, the tool applies both quality measures, *mean average precision* and *discernability*.

The remainder of this paper is organized as follows: Section 2 introduces the two measures for evaluating similarity functions; Section 3 presents the tool and Section 4 presents the conclusions.

2 Evaluating Similarity Functions

The evaluation of similarity functions can be done using different methods. The next subsections present the two measures used in our tool *mean average precision* and *discernability*. Both measures are based on the concept of relevance. An item from the database that represents the same real world object as the one in the query is considered relevant. Conversely, an item from the database that does not represent the same real world entity as the query is considered irrelevant.

In order to calculate both evaluation measures, some resources are necessary: (i) a set of database items to be used as queries; and (ii) a set of similarity functions to be evaluated. Next, a ranking is generated for each similarity function and each query. The ranking is sorted decreasingly by similarity score. Following, a domain expert performs the relevance judgements, i.e. marks the relevant and irrelevant data items on each ranking. An example of a ranking generated by a similarity function is given in Table 1.

2.1 Calculating Mean Average Precision

For each position i in the ranking that corresponds to a relevant item, the precision at the $i-th$ is the ratio $r_i = n/i$, where n is the number of relevant results retrieved up to line i. After calculating the ratio r for all m relevant ranks, the average precision for a query is defined as the average of all m ratios r. The mean average precision (MAP) for a similarity function is the arithmetic mean of the average precisions for the individual queries.

2.2 Calculating *Discernability*

Discernability is a measure specifically designed for evaluating similarity functions proposed by (da Silva et al. 2007). Besides providing a means for evaluating similarity functions, this technique also estimates the optimal threshold t to be used by a similarity function for a data set. This threshold aims at minimizing false negatives and false positives retrieved in response to a set of queries. Details of the *discernability* computation are given in da Silva et al. (2007). This section provides a brief description of the method.

The calculation of *discernability* takes two aspects into consideration:

(i) whether the similarity function succeeded in separating relevant and irrelevant data items. A good similarity function should assign higher scores to all relevant data items than to the irrelevant ones; and

(ii) the level of separation between relevant and irrelevant data items. An ideal similarity function should not only separate relevant and irrelevant data items, but it should also place them within a reasonable distance, creating two clearly distinct sets.

The formula for calculating the *discernability* of a similarity function is given in equation 1:

$$discernability^{L}(t_{best}^{\min}, t_{best}^{\max}, f_{\max}) = \frac{c_1}{c1+c2}(t_{best}^{\max} - t_{best}^{\min}) + \frac{c_2}{c1+c2} \cdot \frac{f_{\max}}{2n} \quad (1)$$

where: L is the similarity function being analysed; $t_{best}^{\min}$ and $t_{best}^{\max}$ are the limits for the optimal threshold interval; c_1 and c_2 are coefficients which allow the user to express the importance given to each of the two aspects considered above; f_{max}, which is explained more thoroughly below, is the number of points achieved by the threshold interval $[t_{best}^{\min}, t_{best}^{\max}]$; and n is the number of queries.

The relevance judgements provided by the user enable the identification of two important points in a ranking generated by a similarity function in response to a query:

- s_{rel} - The lowest score achieved by a relevant data item
- s_{irrel} - The highest score achieved by an irrelevant data item

Example: Consider a database containing titles of Computing Science subjects. The object "Ranking in Databases" is represented in five different forms, namely: "Ranking in Databases", "Ranking on Databases","Ranking on DBs", "Rankin in DBs", "Ranking and DBs". Assuming that the database contains eight data items, a similarity ranking according to the Levenshtein (Levenshtein 1966) metric is given in Table 1. According to this ranking, the lowest score of a relevant item is $s_{rel} = 0.6111$ and the highest score of an irrelevant item is $s_{irrel} = 0.5789$.

Table 1: Example of similarity ranking

Score	Data Item	Relevance
1.0000	Ranking in Databases	Relevant
0.9444	Ranking on Databases	Relevant
0.6111	Ranking on DBs	Relevant
0.6111	Ranking and DBs	Relevant
0.6111	Rankin in DBs	Relevant
0.5789	Relational Databases	Irrelevant
0.4444	Ranking on IR	Irrelevant
0.3889	Ranking Correlation	Irrelevant

In this example, the similarity function has succeeded in separating relevant and irrelevant items since the last relevant item was retrieved before the first irrelevant one. However, the two sets are quite close in the ranking, which is not desirable. When a function fails to do a correct separation, s_{irrel} will be greater than s_{rel}. As a result, the function will be penalized scoring a low *discernability* value.

Plotting a set of queries with their respective s_{rel} and s_{irrel}, as in Figure 1, it is possible to visualize the distance between the relevant and irrelevant data items assigned by the similarity function. In our approach, such a distance is an important parameter used to evaluate the quality of a given similarity metric. As mentioned before, a good similarity function will clearly separate the relevant set from the irrelevant set.

The $BestThresh$ algorithm (da Silva et al. 2007), which finds the optimal threshold interval (highlighted gray in Figure 1), is based on a reward function. It proceeds as follows: Each threshold t in the interval [0,1] (varying according to a predefined numeric precision), is compared to s_{rel} and s_{irrel} for the rankings produced in response to a number of queries. One of three outcomes is possible from such comparisons:

(i) the threshold t is at the same time less than s_{rel} and greater than s_{irrel}. This means that it is able to separate relevant and irrelevant items, so it earns two points.

(ii) the threshold t satisfies only one of the conditions. This means that both relevant and irrelevant items are on the same side (either above or below) the line drawn by the threshold. It then scores zero points.

(iii) the threshold t fails both conditions. In that situation, the last relevant result is below the threshold line whilst the first irrelevant result is above it. As a result, t loses 2 points.

The algorithm then searches for the highest number of points (f_{max}) achieved by a threshold. Once f_{max} is found, the algorithm searches for the contiguous interval of values of t ($[t_{best}^{\min}, t_{best}^{\max}]$) that achieve ($f_{max}$).

In addition to $BestThresh$, da Silva et al. (2007) propose a statistical method for finding the optimal threshold. This method is based on the distribution of s_{rel} and s_{irrel} values for a sample of n queries. Experimental results show that both methods for threshold estimation are in agreement. For the tests performed in this paper we have used the algorithmic method $BestThresh$.

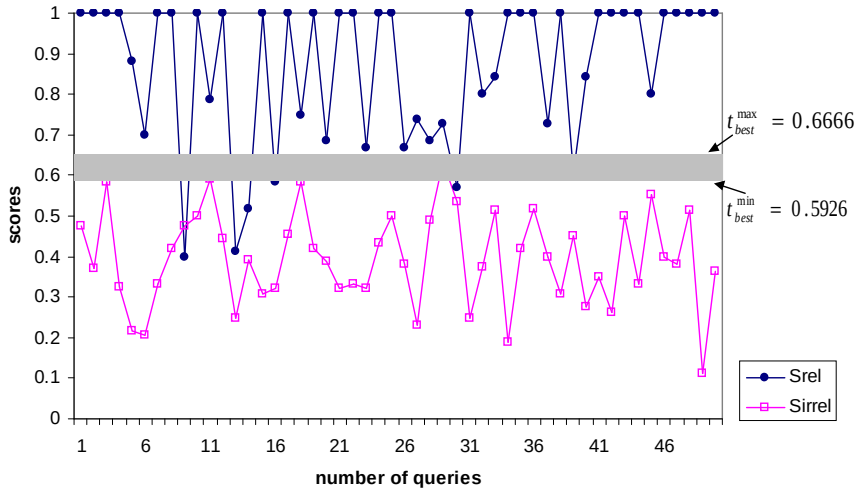

Figure 1: Plot of s_{rel} (dots) and s_{irrel} (squares) values over 50 queries. The interval of threshold that best separates relevant and irrelevant data items is highlighted.

2.3 Comparing *MAP* and *Discernability*

Comparing both evaluation measures it was possible to conclude that *discernability* is able to find differences between similarity functions that are considered identical by *MAP*. This is due to the fact that *discernability* takes the scores assigned by the similarity functions to relevant and irrelevant data items into consideration and not just their ranks. In addition, the two measures not always agree on the best similarity function. These conflicts do not mean that one of the measures is wrong in its judgement, since they were designed for different purposes. *MAP* was designed to assess rankings produced in response to IR-style queries. On the other hand, *discernability* was designed to assess rankings produced in response to range queries.

3 The *SimEval* Tool

The *SimEval* tool, described in this paper, aims at assessing the quality of similarity functions applied to a given domain. *SimEval* is a web application, developed in Java, which uses the MySQL database and a TomCat server. The URL for accessing the tool is `http://www.inf.ufrgs.br/~heuser/simeval.html`.

The input to the tool is a file with the dataset for which the evaluation will be performed. The format for the data file is the same used by FEBRL (Freely Extensible Biomedical Record Linkage) proposed by Christen et al. (2004). FEBRL generates synthetic data sets to be used by record linkage and deduplication systems. There are two main advantages of adopting this format:

(i) The relevance judgements are provided within the file. The identification of each data instance can be processed to provide relevance information.

(ii) Synthetic data sets produced with FEBRL can be directly used by the tool. This facilitates the running of experiments.

An example of a fragment from the file is shown in Figure 2.

The current version of the tool is designed to evaluate data sets containing a single column. Therefore,

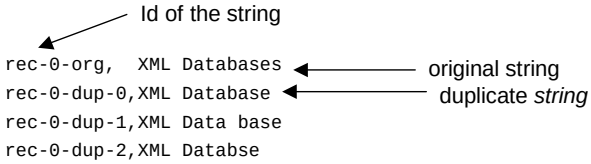

Figure 5: Input file format

if the input data file contains a set of attributes (tuples), those will be concatenated into a single string and processed by the tool.

The *SimEval* tool provides all 23 similarity functions available from SimMetrics (Chapman 2007), which is an open source library of similarity functions. It includes a range of similarity measures from a variety of communities, including statistics, DNA analysis, artificial intelligence, information retrieval, and databases. These are listed below:

- Soundex
- JaccardSimilarity
- SmithWaterman
- JaroWinkler
- SmithWatermanGotoh
- EuclideanDistance
- TagLinkToken
- ChapmanMeanLength
- QGramsDistance
- ChapmanMatchingSoundex
- CosineSimilarity
- OverlapCoefficient
- Levenshtein
- Jaro
- NeedlemanWunch
- MongeElkan
- TagLink

- DiceSimilarity
- SmithWatermanGotohWindowedAffine
- ChapmanOrderedNameCompoundSimilarity
- BlockDistance
- ChapmanLengthDeviation
- MatchingCoefficient

The tool allows the generation of several samples of queries, for specific similarity functions. For example, it is possible to process 10 samples of 50 queries each, using Soundex, JaroWinkler and Levenshtein.

In order to visualize the results of the evaluation measures, the tool displays the scores calculated by *discernability* and MAP for all samples. It is also possible to visualize the details for a sample, or the ranking for a specific query.

The tool is very easy to use. Just three steps are necessary:

1. Upload data file: the user informs the file containing the data set in FEBRL format. This step is shown in Figure 3.
2. Sample generation: the user chooses which similarity functions to use, how many samples and how many queries in each sample. This step is shown in Figure 4.
3. Analysis of results: the scores for MAP and *discernability* are shown to the user. All results are stored in a MySQL database, thus it is possible to see results previously generated. This step is shown in Figure 5.

The screen that displays the results has a cell for each sample. A click on the cell opens the screen with the details of the sample. This screen shows, for example, the values of s_{rel} and s_{irrel} for each query. By clicking on a cell with the query results, the user is taken to a screen that displays the ranking of all database items in response to that query.

4 Conclusions

In order to perform range queries it is necessary to evaluate which similarity function is most suitable for a given application. The *SimEval* tool, presented in this paper, aims at fulfilling this need using two evaluation measures *discernability* and MAP.

Amongst the possibilities for improvements there are performance enhancements that can be achieved through the implementation of the similarity functions directly into the database, and the generation of graphics to aid the visualization of the results.

References

Benjelloun, O., Garcia-Molina, H., Kawai, H., Larson, T. E., Menestrina, D., Su, Q., Thavisomboon, S. & Widom, J. (2006), 'Generic entity resolution in the serf project', *IEEE Data Engineering Bulletin* **June**.

Chapman, S. (2007), 'Simmetrics', `http://www.dcs.shef.ac.uk/~sam/simmetrics.html`.

Christen, P., Churches, T. & Hegland, M. (2004), Febrl - a parallel open source data linkage system, *in* 'Proceedings of the 8th Pacific-Asia Conference, PAKDD (LNAI 3056)', Springer, pp. 638–647.

Cohen, W. W. (2000), 'Data integration using similarity joins and a word-based information representation language.', *ACM Trans. Inf. Syst.* **18**(3), 288–321.

da Silva, R., Stasiu, R. K., Orengo, V. M. & Heuser, C. A. (2007), 'Measuring quality of similarity functions in approximate data matching', *Journal of Informetrics* **1**(1), 35–46. doi:10.1016/j.joi.2006.09.001.

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 6th annual ACM international workshop on Web information and data management', ACM Press, New York, NY, USA, pp. 56–63.

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

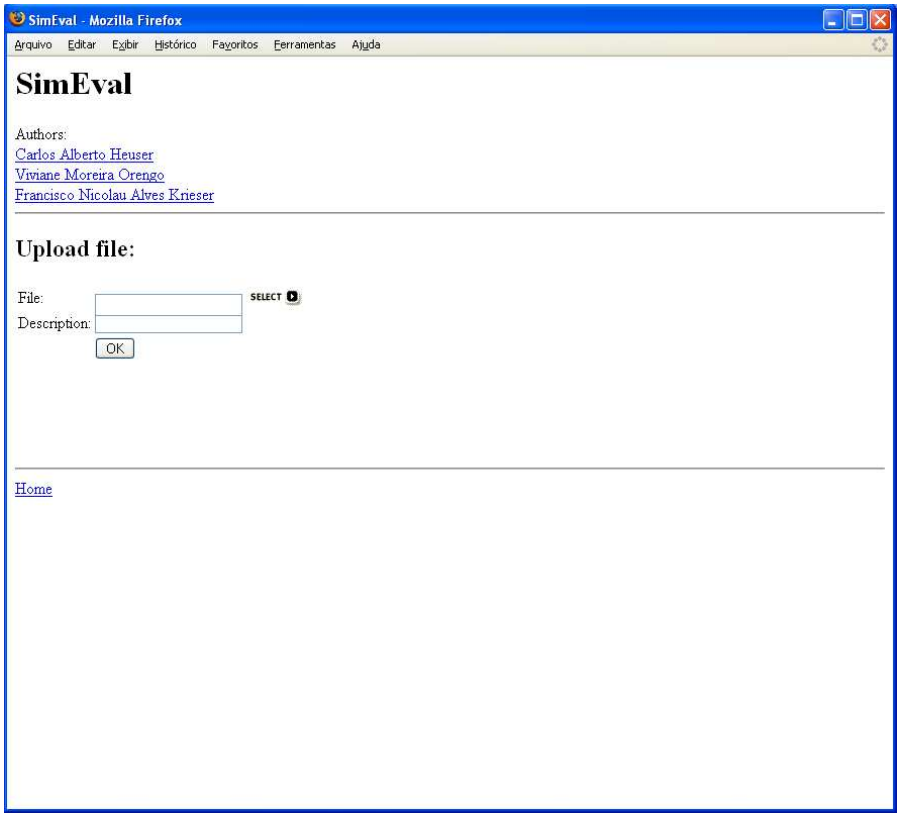

Figure 2: Step 1 - Upload data file

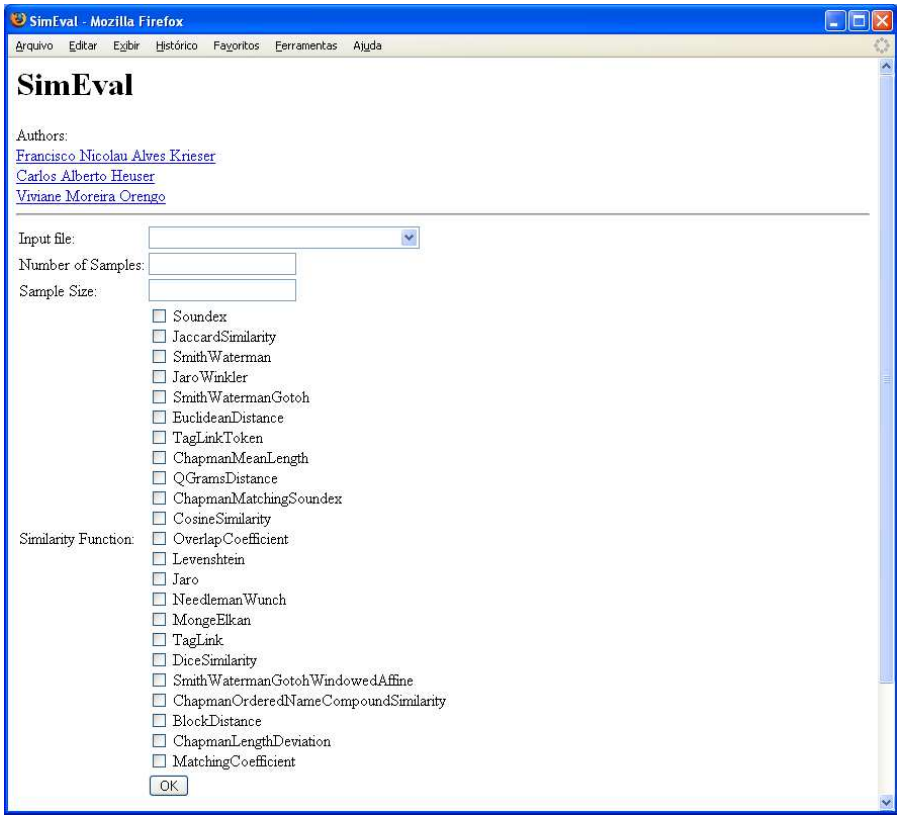

Figure 3: Step 2 - Choose sample size, number of samples and similarity functions

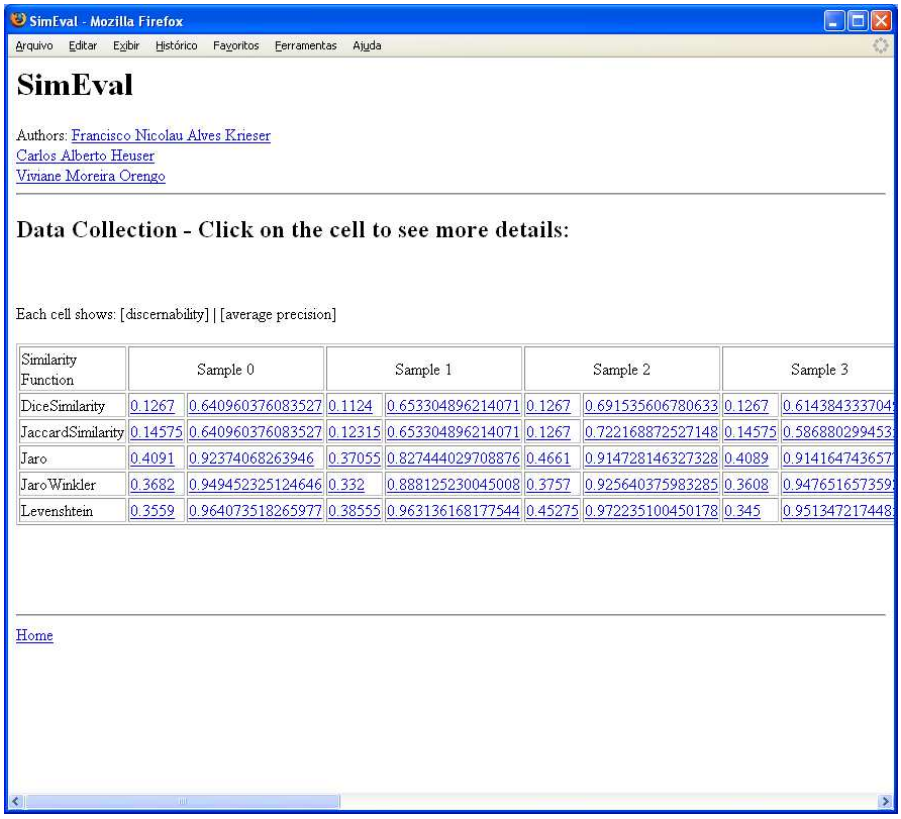

Figure 4: Step 3 - Analyze results. The scores for *discernability* and MAP are shown for each sample and each similarity function