Automatic Threshold Estimation for Data Matching Applications

Juliana Bonato dos Santos, Carlos A. Heuser, Viviane Moreira Orengo, Leandro Krug Wives

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
{jbonato,heuser,vmorengo,wives}@inf.ufrgs.br

Abstract. Several advanced data management applications, such as data integration, data deduplication or similarity querying rely on the application of similarity functions. A similarity function requires the definition of a threshold value in order to assess if two different data instances match, i.e., if they represent the same real world object. In this context, the threshold definition is a central problem. In this paper, we propose a method for the estimation of the quality of a similarity function. Quality is measured in terms of recall and precision calculated at several different thresholds. On the basis of the results of the proposed estimation process, and taking into account the requirements of a specific application, a user is able to choose a threshold value that is adequate for the application. The proposed estimation process is based on a clustering phase performed on a sample taken from a data collection and requires no human intervention.

1 Introduction

Several data matching applications, like data integration, data deduplication, data cleaning, similarity querying and similarity joining, employ similarity functions as the central resource to assess whether or not two different data instances represent the same real world object. Given two data instances (e.g. “VLDB” and “Very Large Databases”), a similarity function returns a similarity score. If this similarity score is higher than a predefined threshold, we say that the data instances match, i.e. the data instances are considered to represent the same real world object. Notice that data instances are not restricted to simple strings as in the example above. They can also be aggregate data instances, such as tuples or XML trees.

A shortcoming of similarity functions is that their score values are not usually meaningful. For instance, a specific threshold, say 0.7, may lead to high precision with one similarity function and low precision with another. This means that in order to choose the threshold for a specific application, some knowledge about the quality of the similarity function at different thresholds must be available. If the dataset to be matched or a sample thereof is available, this knowledge may be obtained by the following estimation process:

1. Take samples from the dataset that contains the instances to be matched by the similarity function;
2. Pick some data instances from the sample (these instances are called query values);
3. For each query value, evaluate the sample, i.e., label the items from the sample as relevant or irrelevant depending on whether or not they match the query value;
4. Compare each query value with all elements in the sample and compute precision and recall figures at several different score values;
5. Compute the average precision and recall at different score values over all queries taken in Step 3.

This procedure results in a table that maps each of the 11 threshold values to an estimation of recall and precision at these values. Having obtained this table and taking the requirements of the specific application (high precision, high recall or best combination of both) into account, one is able to choose an adequate threshold for the application at hand.

However, this procedure is highly dependent on human intervention. In Step 3, a human expert must take each query value and compare it to each other value in the sample and decide if the value matches the query (relevant value) or if it does not (irrelevant value).

In previous work, the authors propose a procedure that aims at reducing this human intervention [Stasiu et al. 2005]. In that procedure, instead of labeling relevant and irrelevant values, the human expert just informs how many different real world objects are represented by the values in samples taken from the dataset to be processed. For example, in an application that matches strings representing conference names, the human expert is presented with a sample of different conference names and informs how many different conferences are represented by these strings. In this example, the strings “VLDB”, “Intl. Conf. on Very Large Databases” and “Very Large Databases” would be counted just once, as they refer to a single conference. Experiments show the feasibility of this approach with three or four samples of 40~50 data instances each. A limitation of this study is that the sample size was influenced by the human’s ability in counting the number of elements in the sample. The authors found that 50 items were the maximum the experts could deal with.

In the present paper, we take a step further and eliminate human intervention completely. As demonstrated by our experiments, if a data sample is clustered in such a way that the silhouette coefficient, a quality measure for clustering algorithms [Kaufman and Rousseeuw 1990] is maximized, each cluster tends to contain instances that represent a single real world object. On the basis of the results of this clustering process, we are able to estimate precision and recall values for different thresholds. Notice that, the silhouette coefficient is an internal evaluation measure [Manning et al. 2008]. Internal evaluation measures do not require an evaluated dataset, i.e., does not require that matching data instances are previously known. Thus, our approach does not require human intervention. By eliminating human intervention in the estimation process, we are able to handle much larger data samples. This potentially leads to more precise estimations.

The experiments presented in this paper show that the precision and recall values estimated by the proposed procedure hold with a small error margin for larger datasets from the same domain.

The paper is organized as follows. Section 2 reviews some related work. In Section 3, we first shortly present the procedure we have previously developed [Stasiu et al. 2005] and then detail our approach. Section 4 presents the experiments that empirically demonstrate that (i) the results of our estimation process are close to those obtained when a
human expert labels the samples as relevant and irrelevant values, and (ii) the recall and precision values estimated by using the proposed procedure hold for the entire dataset. Section 5 presents the concluding remarks and outlines some future work.

2 Related Work

Approximate data matching is a core operation in data management applications like flexible query processing, data deduplication and data cleaning.

In flexible query processing, the goal is to retrieve data instances even when users have a limited understanding of the data contents. In this case, instead of requiring values in tuples to exactly match the arguments in the queries, approximate matching is used [Motro 1988; Dalvi and Suciu 2004]. For this purpose, similarity functions based on characters (e.g., Levenshtein distance, n-gram distances), tokens (e.g., Jaccard distance, TF/IDF), phonetics (e.g., soundex distance) or semantics (e.g., ontological distance) are used to match values to query arguments and to assign a similarity score to evaluate the match. The degree of similarity between query arguments and database values is usually bound by a user-specified threshold.

Approximate matching has also been widely employed in data integration applications, since the seminal work on the so called record linkage or data deduplication problem [Fellegi and Sunter 1969]. In this work, the similarity between records of two distinct files is given by the sum of the similarity scores between pairs of common attributes (fields) of these files. A special case is the approximate join operator [Gravano et al. 2003; Guha et al. 2006] that matches records from different files according to the degree of similarity between their fields.

A further application of data matching is data cleaning [Chaudhuri et al. 2003; Guha et al. 2006]. In this application, data matching is used to detect and eliminate errors that appear in data received at a data warehouse from external sources.

All these applications have two aspects in common: (i) the use of similarity functions and (ii) the problem of specifying a proper threshold. In order to define the threshold, the user must somehow understand the meaning of the scores returned by the similarity function. One way to express this meaning is to estimate the quality of the results of a similarity function at a specific threshold value. For this purpose, classical Information Retrieval quality measures like recall, precision and F-measure are usually employed [Baeza-Yates and Ribeiro-Neto 1999]. Such measures are sometimes called external quality measures [Manning et al. 2008], as they require human intervention to analyze the data.

Clustering is a process used to identify clusters (i.e., groups) of relatively homogeneous elements. Clusters are created based on the on the similarities and relationships amongst the data items themselves [Kaufman and Rousseeuw 1990]. In this area, internal quality measures [Manning et al. 2008], i.e., quality measures that do not require human intervention, were developed.

Cohesion is an internal quality measure that expresses the average similarity between the elements of a cluster. The larger the similarity amongst the elements of a
group, the more cohesive the group is. The cohesion of a cluster $C$ can be calculated as shown in Equation 1 [Kunz and Black 1995].

$$cohesion(C) = \frac{\sum_{i,j} \text{sim}(c_i, c_j)}{m(m-1)/2}$$

Equation (1)

where $\text{sim}(c_i, c_j)$ is the similarity score between the elements $c_i$ and $c_j$ belonging to cluster $C$, and $m$ is the number of elements in cluster $C$.

Coupling is an internal quality measure that expresses the average similarity between all pairs of elements in which one element belongs to cluster $C$ and the other does not. Ideally, coupling should be low. Coupling is calculated as in Equation 2 [Kunz and Black 1995]:

$$coupling(C) = \frac{\sum_{i,j} \text{sim}(c_i, q_j)}{m \times n}$$

Equation(2)

where $\text{sim}(c_i, q_j)$ is the similarity between the element $c_i$ from cluster $C$ and element $q_j$ from another cluster; $m$ is the number of elements in $C$ and $n$ is the number of elements outside $C$.

The silhouette coefficient is another internal quality measure that combines cohesion and coupling, and can be used to evaluate the effectiveness of clustering results. In our case, it was used to identify the most appropriate cluster and to avoid human intervention in the process. According to [Tan et al. 2006; Aranganayagi and Thangavel 2007] the silhouette coefficient of the $i^{th}$ element can be calculated as in Equation 3:

$$SilhouetteCoefficient(i) = \frac{b_i - a_i}{\max(a_i, b_i)}$$

Equation(3)

where, $a_i$ is the average dissimilarity (i.e., distance) between the $i^{th}$ element of the cluster and the other objects of the same cluster; and $b_i$ is the minimum average dissimilarity between the $i^{th}$ element and any cluster not containing the element. In our case, we use dissimilarity to calculate the silhouette coefficient, but an analogous approach can be used for similarities. The silhouette coefficient of a cluster is the average of the coefficients calculated for its elements.

Recently, some work has dealt specifically with the problem of threshold specification. As already mentioned in the introduction, the authors have developed a method that reduces human intervention in the process of estimation of the quality of a similarity function. In another work [da Silva et al. 2007], the “best” threshold for a similarity function is obtained as a byproduct of the process in which discernability is calculated. Discernability is a quality measure designed specifically for similarity functions. In that work, “best” threshold means that by employing this threshold one will maximize recall and precision at the same time. However, this threshold may not be adequate for every data matching application, as each application may have special requirements in terms of recall and precision.
3 Quality estimation for similarity functions

In this section we present background work in which our proposed approach is based [Stasiu et al. 2005] and then introduce our approach.

3.1 Semi-automatic Approach

As already mentioned, a central problem in the application of similarity functions is the threshold definition. In order to choose a threshold value for a specific application, knowledge about the quality of the results achieved by the similarity function is needed. The quality of similarity functions is usually measured in terms of recall and precision values [Baeza-Yates and Ribeiro-Neto 1999]. If recall and precision for different threshold values are known, one is able to choose an adequate threshold value for the specific application. This knowledge about the quality of a similarity function may be acquired by the estimation process described in the introduction, given that a sample of the data to be matched by the similarity function is provided. Notice that this estimation process relies heavily on human intervention, as some human expert has to manually identify relevant and irrelevant matches.

Aiming at reducing human intervention, in previous work [Stasiu et al. 2005], the authors propose an estimation procedure in which, instead of labeling relevant and irrelevant values, the human expert just informs how many different real world objects are represented by the values in samples of the dataset to be processed. Specifically, the estimation procedure proposed in the aforementioned work is the following:

1. **Sampling**

   A random sample of values in the dataset is taken. Experiments have shown that four samples of 40~50 data instances each are adequate, i.e. provide an estimation error of up to 10%.

2. **Human intervention**

   The values in the sample are shown to a human expert. The human expert then counts the number of different real world objects that are represented by the data instances in the sample. Remember that different values may represent the same real world object.

3. **Clustering**

   The values in the sample are clustered in such a way that the number of clusters best approaches the number of different real world objects identified by the human expert. The underlying idea is that, if the similarity metric behaves correctly, each cluster will contain data instances that represent a single real world object. Here is important to state that even if the number clusters found by the clustering algorithm is the same as the number of distinct objects, we cannot say that the clustering process is correct as it is possible that a cluster contains more than one real world object and/or values corresponding to the same object were partitioned across different groups. However, experiments reported on [Stasiu et al. 2005] show that the error margin is small.

4. **Recall/precision estimation**

   Considering that each cluster contains only matching data instances, i.e. values that represent the same real world object, precision and recall at 11 different threshold values (0,
0.1, 0.2, ..., 1) are computed by the classical Information Retrieval approach [Baeza-Yates and Ribeiro-Neto 1999]. Recall is the fraction of relevant data items that were retrieved and precision is the fraction of the retrieved items that are indeed relevant.

Experiments reported in [Stasiu et al. 2005] demonstrate that precision and recall values estimated for the data samples hold with a small error margin for the entire dataset.

3.2 Automatic Approach

In this paper, we propose a method for eliminating human intervention in the process of recall/precision estimation. As our experiments demonstrate, when data instances are clustered in such a way that each cluster contains only instances that match, the silhouette coefficient [Tan et al. 2006; Aranganayagi and Thangavel 2007], a quality measure for clusters, is maximized. Thus, instead of requiring human intervention as in the process developed previously, we cluster each sample of data instances using different thresholds and choose the cluster configuration that achieves the highest silhouette coefficient. For details on the calculation of the silhouette coefficient, please refer to Section 2.

The proposed estimation process, now without human intervention, is as follows:

1. Clustering

The values in the dataset are clustered in such a way that the silhouette coefficient is maximized. The underlying idea is that, if the similarity metric behaves correctly, each cluster will contain data instances that represent a single real world object. Notice that if the dataset that is available for estimation is very large, a sample thereof could be taken.

2. Recall/precision estimation (same as Step 4 in the semi-automatic approach described in Section 3.1).

In order to estimate recall and precision, we use each value in the dataset as a query object. The values in the dataset are then ranked according to their similarity to this query object and recall and precision are calculated by the classical Information Retrieval process [Baeza-Yates and Ribeiro-Neto 1999] at 11 different threshold values (0, 0.1, 0.2, ..., 1). In this computation, we consider that the data instances contained in a cluster match, i.e., we consider the values contained in the same cluster as the query object as the relevant answers to this query. This procedure is repeated for each value in the dataset. At last, the average of recall and precision at the 11 thresholds over the entire dataset is taken.

The result is a table that maps each of the 11 different threshold values to the estimated recall and precision at these thresholds. Table 1 has an example of such table. As expected, as the threshold increases, there is a decrease in recall and a rise in precision.
Table 1 – Recall and Precision values at different thresholds

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1.0000</td>
<td>0.2566</td>
</tr>
<tr>
<td>0.2</td>
<td>0.9989</td>
<td>0.3937</td>
</tr>
<tr>
<td>0.3</td>
<td>0.9868</td>
<td>0.5851</td>
</tr>
<tr>
<td>0.4</td>
<td>0.8910</td>
<td>0.8701</td>
</tr>
<tr>
<td>0.5</td>
<td>0.8257</td>
<td>0.9807</td>
</tr>
<tr>
<td>0.6</td>
<td>0.7837</td>
<td>1.0000</td>
</tr>
<tr>
<td>0.7</td>
<td>0.6396</td>
<td>1.0000</td>
</tr>
<tr>
<td>0.8</td>
<td>0.3927</td>
<td>1.0000</td>
</tr>
<tr>
<td>0.9</td>
<td>0.1069</td>
<td>1.0000</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0351</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

4 Experiments and evaluation

In this section, the experiments performed to empirically validate our proposed method are described. The experiments have two goals.

The first goal is to demonstrate that by using the threshold value that maximizes the silhouette coefficient, we achieve clusters that correctly group matching data instances. Thus, we are demonstrating that we do not require user intervention in order to cluster together matching data instances.

The second goal is to demonstrate that recall and precision values estimated by our procedure also hold for a larger dataset, with a small margin of error.

4.1 Datasets

We have generated synthetic datasets using FEBRL (Freely Extensible Biomedical Record Linkage) [Christen et al. 2004]. FEBRL generates artificial datasets to be used by record linkage and deduplication systems. The reasons for using artificial data collections instead of real ones are twofold: (i) we know the gold standard classes, i.e. the dataset is already evaluated, and thus we are able to compute external quality measures like recall and precision; and (ii) we are able to control several parameters such as the number of original and duplicate records.

Table 2 – Examples of records from the datasets

<table>
<thead>
<tr>
<th>rec_id</th>
<th>name</th>
<th>rec_id</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>rec-1-org</td>
<td>kyle ryan</td>
<td>rec-2-org</td>
<td>nicholas akot</td>
</tr>
<tr>
<td>rec-1-dup-1</td>
<td>kyle ryah</td>
<td>rec-2-dup-1</td>
<td>nichlas akot</td>
</tr>
<tr>
<td>rec-1-dup-2</td>
<td>kile ryan</td>
<td>rec-2-dup-2</td>
<td>nicholas aokt</td>
</tr>
</tbody>
</table>
Table 2 shows a fragment from one of the datasets. The first column contains the identifier of the data item. From this annotation, we know to which cluster the element belongs and that enables us to calculate external validity measures such as recall, precision and F-measure.

Four datasets were generated for the estimation process. The average number of elements in the cluster was varied systematically to enable the comparison of the quality metrics for datasets with smaller and larger clusters. The domain of the data values is person names. Table 3 shows the details of the estimation datasets.

For test purposes, in order to evaluate if the result of the estimation process holds for larger datasets, we have generated another set of four datasets. These datasets have ten times the number of records contained in the datasets used in the estimation process. The average number of items per cluster remains the same as in the estimation datasets. Table 4 shows the details of the test datasets.

### Table 3 – Details of the estimation datasets

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>e_names1</th>
<th>e_names2</th>
<th>e_names3</th>
<th>e_names4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of distinct objects</td>
<td>100</td>
<td>30</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>Total number of records</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Average number of objects per cluster</td>
<td>2</td>
<td>6.67</td>
<td>13.33</td>
<td>27.57</td>
</tr>
</tbody>
</table>

### Table 4 – Details of the test datasets

<table>
<thead>
<tr>
<th>Dataset Name</th>
<th>t_names1</th>
<th>t_names2</th>
<th>t_names3</th>
<th>t_names4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of distinct objects</td>
<td>1000</td>
<td>300</td>
<td>150</td>
<td>70</td>
</tr>
<tr>
<td>Total number of records</td>
<td>2000</td>
<td>2000</td>
<td>2000</td>
<td>2000</td>
</tr>
<tr>
<td>Average number of objects per cluster</td>
<td>2</td>
<td>6.67</td>
<td>13.33</td>
<td>27.57</td>
</tr>
</tbody>
</table>

#### 4.2 Choice of Similarity Functions and Clustering Algorithms

The literature contains a large variety of similarity functions created for different purposes, e.g. string matching, DNA matching or tuple matching. The choice of the similarity function is crucial to the quality of clustering process. Thus, we should choose adequate metrics for the domain in question. In order to choose the most suitable functions, we have used the SimEval tool [Heuser et al. 2007]. SimEval implements 23 similarity functions and evaluates them in terms of Mean Average Precision, which is the standard evaluation metric for Information Retrieval, and Discernability [da Silva et al. 2007], which is a metric specifically designed for evaluating similarity functions. The results show that the most suitable similarity functions for our datasets are: Levenshtein [Levenshtein 1966] SmithWaterman [Smith and Waterman 1981], QGrams [Gravano et al. 2001] and Carla [Mergen and Heuser 2005].

The datasets were clustered using four hierarchical agglomerative clustering algorithms, namely: SingleLink [Sneath and Sokal 1973], CompleteLink [King 1967],
Group-Average [Aldenderfer and Blashfield 1984] and Ward [Ward 1963]. Their performances in terms of time and quality were vastly similar. Therefore, the results reported here are for the Group-Average algorithm which achieved slightly better results.

4.3 Procedure

The first set of experiments aims at demonstrating that by using the threshold value that maximizes the silhouette coefficient we achieve clusters that correctly group matching data instances. Our goal here is to find correlations between the quality of the clusters as measured by the silhouette coefficient and the quality of the clusters by the F-measure. The F-measure is the weighted harmonic mean between recall and precision. We have used equal weights for both. Bear in mind that the silhouette coefficient is an internal quality measure [Manning et al. 2008], i.e. a measure that does not rely on a previous definition of the correct clustering, whereas the F-measure is an external quality measure, i.e. requires the dataset to be previously evaluated. Recall further that the datasets generated by FEBRL are already evaluated, i.e., they indicate the correct clustering. Thus, in order to calculate the F-measure, we used the gold-standard classes from FEBRL. However, for the silhouette coefficient we considered clustering information only.

For this experiment, each dataset was clustered using four distinct similarity functions (see section 4.2). For each similarity function, eleven different thresholds were used (0.0, 0.1, 0.2, …, 1). For the 44 runs, we computed the silhouette coefficient and the F-measure. In order to compare silhouette coefficient and F-measure values the correlation between these values was computed for each dataset and similarity function.

The second set of experiments aims at demonstrating that the mapping between thresholds and recall/precision values estimated by our procedure also holds for larger datasets. For this experiment, we compared the estimated recall and precision values obtained by our process on the estimation datasets with recall and precision values computed on the larger test datasets. The values were compared by computing the Mean Square Deviation (MSD) between recall/precision estimated values and recall/precision values computed on the test datasets. MSD is a widely used measure in statistics to assess how much an estimation differs from the actual value being estimated. MSD is calculated as in eq. 4.

\[
MSD = \frac{1}{n} \sum_{j=1}^{n} (\hat{x}_j - x_j)^2
\]  

Equation(4)

where \(n\) is the number of threshold values analyzed, \(\hat{x}\) refers to the values from the estimation dataset and \(x\) refers to the values from the test dataset.

4.4 Results

The results of the first set of experiments show that silhouette coefficient is highly correlated with F-measure. Figure 1 shows four graphs, one for each dataset, containing the values of F-measure and silhouette coefficient for 11 thresholds. These graphs refer to the QGrams similarity function. The graphs for the remaining three similarity functions are not depicted in the paper for space reasons, but they show similar levels of correlation between F-measure and silhouette coefficient. The graphs show that in several cases the threshold that maximizes F-measure is the same threshold that maximizes the silhouette coefficient.
This agreement occurred 9 out of 16 times (4 datasets times 4 similarity functions), revealing that it can be used as a reliable criteria of quality to enable the choice of a threshold.

Figure 1 - Values for F-Measure and Silhouette Coefficient for 11 thresholds – Qgrams similarity function and Group-Average clustering algorithm

Table 5 – Correlation between F-Measure and Silhouette Coefficient

<table>
<thead>
<tr>
<th></th>
<th>Levenshtein</th>
<th>Carla</th>
<th>Q-Grams</th>
<th>Smith-Watermann</th>
</tr>
</thead>
<tbody>
<tr>
<td>e_Names1</td>
<td>0.482</td>
<td>0.604</td>
<td>0.757</td>
<td>0.770</td>
</tr>
<tr>
<td>e_Names2</td>
<td>0.977</td>
<td>0.975</td>
<td>0.985</td>
<td>0.986</td>
</tr>
<tr>
<td>e_Names3</td>
<td>0.983</td>
<td>0.991</td>
<td>0.981</td>
<td>0.971</td>
</tr>
<tr>
<td>e_Names4</td>
<td>0.926</td>
<td>0.942</td>
<td>0.970</td>
<td>0.942</td>
</tr>
</tbody>
</table>
Table 5 shows the correlation coefficient between F-measure and silhouette coefficient for all four datasets and all similarity functions. The table shows high values of correlation (correlation coefficient above 92%) for datasets e_Names2-4.

However, in the case of the e_Names1 dataset, the correlation coefficient is lower for all four similarity functions. The reason for this is that the Names1 dataset contains relatively small clusters. On the average, there are two different data instances representing one real world object, i.e., the average cluster size is two. This means that for this specific dataset a similarity function must be very precise and every incorrect match has a relatively high impact on the quality measures.

The results of the second set of experiments demonstrate that recall and precision values estimated by our process hold with a small error margin for a bigger dataset. Figure 2 shows values of recall and precision at different threshold values for the Carla similarity function and for the four datasets. Estimated recall and precision values are those calculated by our procedure using the estimation datasets (see Section 4.1) and test recall and precision values are those computed on the test datasets. Due to space reasons again we show the result for a single similarity function (QGrains).

<table>
<thead>
<tr>
<th></th>
<th>Levenshtein</th>
<th>Carla</th>
<th>Q-Grains</th>
<th>Smith-Watermann</th>
</tr>
</thead>
<tbody>
<tr>
<td>Names1</td>
<td>0.0021</td>
<td>0.0147</td>
<td>0.0154</td>
<td>0.0056</td>
</tr>
<tr>
<td>Names2</td>
<td>0.0001</td>
<td>0.0000</td>
<td>0.0036</td>
<td>0.0001</td>
</tr>
<tr>
<td>Names3</td>
<td>0.0007</td>
<td>0.0044</td>
<td>0.0043</td>
<td>0.0252</td>
</tr>
<tr>
<td>Names4</td>
<td>0.0058</td>
<td>0.0059</td>
<td>0.0047</td>
<td>0.0029</td>
</tr>
</tbody>
</table>

Table 6 shows MSD values obtained by comparing estimated recall with test recall and Table 7 shows the MSD values estimated precision with test precision.

Table 6 – Mean Square Deviation between estimated recall and test recall

Table 7 – Mean Square Deviation between estimated precision and test precision

Table 6 shows MSD values obtained by comparing estimated recall with test recall and Table 7 shows the MSD values estimated precision with test precision.

As Tables 6 and 7 show, our estimation procedure is very precise in estimating recall values (MSD values below 2.52%) and less precise in estimating precision values (MSD values below 10.42%). These values are comparable to the ones reported by Stasiu et al. 2005. This fact is an indication that the elimination of human intervention did not have a significant impact on the estimation process.
Figure 2 – Recall and precision values for various thresholds (estimated values and test values) QGrams similarity function
5 Conclusion

Threshold definition is an important issue not only in data matching applications but also in other tasks that involve the use of similarity functions. Threshold definition usually relies on an estimation of the effect of the threshold value on the quality of the results of the similarity function. This quality estimation may be performed on a sample of the data to be matched and relies heavily on human intervention by labeling each pair of values as “match” or “no match”.

In this paper, we propose an estimation method that does not require human intervention. By not requiring human intervention the method potentially leads to better results, as larger samples of data may be employed in the estimation process.

Our experiments show the feasibility of the proposed estimation method.

The experiments also show that for some datasets the method leads to less accurate results. These are usually data samples that are “hard” to be handled, either because the sample contains very small clusters or because the similarity function is not adequate for data in the specific domain. We are now working on the problem of identifying in what conditions our method leads to better results. Another problem that must be solved is that of defining the size of the dataset to be used in the estimation process.

References


