A Generic and Flexible Framework for Selecting Correspondences in Matching and Alignment Problems

Fabien Duchateau[1]
[1] *Université Lyon 1, LIRIS, UMR5205, Lyon, France*
fduchate@liris.cnrs.fr

Keywords: Data Integration, Schema Matching, Ontology Alignment, Entity Resolution, Entity Matching, Selection of Correspondences

Abstract: The Web 2.0 and the inexpensive cost of storage have pushed towards an exponential growth in the volume of collected and produced data. However, the integration of distributed and heterogeneous data sources has become the bottleneck for many applications, and it therefore still largely relies on manual tasks. One of this task, named *matching* or *alignment*, is the discovery of correspondences, i.e., semantically-equivalent elements in different data sources. Most approaches which attempt to solve this challenge face the issue of deciding whether a pair of elements is a correspondence or not, given the similarity value(s) computed for this pair. In this paper, we propose a generic and flexible framework for selecting the correspondences by relying on the discriminative similarity values for a pair. Running experiments on a public dataset has demonstrated the improvment in terms of quality and the robustness for adding new similarity measures without user intervention for tuning.

1 INTRODUCTION

Organizations, companies, science labs and Internet users produce a large amount of data everyday. Fusioning catalogs of products, generating new knowledge from scientific databases, helping decision-makers during catastrophic scenarios or creating new mashups are only a few examples of application that involve the integration of distributed and heterogeneous data sources. Unfortunately, the data integration task is still largely performed manually, in a labor-intensive and error-prone process. One of the basic task when integrating data sources deals with the discovery of semantically-equivalent elements, and the link drawn between such elements is a correspondence. Given the nature of the data sources, this task is referred to as **schema matching** (Bellahsene et al., 2011; Bernstein et al., 2011), **ontology alignment** (Euzenat and Shvaiko, 2007; Avesani et al., 2005) or **entity resolution** (Fellegi and Sunter, 1969; Winkler, 2006). An example of schema matching task which occurs during the creation of a mediation system for flight booking is the discovery of correspondences between the Web forms (of the flight companies) and the mediated schema. A similar problem involving ontology alignment may be found in query answering on the Linked Open Data: one needs to understand the relationships between the concepts and properties of the ontologies of the knowledge bases to return complete and minimal query results. In entity resolution, the merging of two databases about products sold by different companies imply the detection of identical products.

To tackle these challenges, matching tools apply a diversity of similarity measures between two elements to exploit the different information stored in the data sources. For most of the tools, the values computed by these similarity measures are finally combined into a global similarity score (Doan et al., 2003; Aumueller et al., 2005; Christen, 2008; Bozovic and Vassalos, 2008). This score can be used to rank and present to the user the top-K candidate correspondences for a given element. A tool can also automatically select the correspondences by comparing this global score with a threshold. In all cases, this task, that we further name **selection of correspondences**, is therefore crucial in the matching process. Yet, we advocate that this global score is sufficient neither for a manual nor for an automatic selection of the correspondences. Indeed, it can reflect the strong impact of similarity measures of the same type if the combination function is not correctly tuned. Besides, the score aggregates all computed similarity values, although most of them may not be significant. Thus, applying a threshold on the global score to select correspondences may not be the best solution because a correct correspondence rarely achieves high values for all similarity measures. Finally, a similarity measure returns similarity scores but we usually do not know its inability for discovering a correspondence. Many measures are applied in a similar fashion, or on the same elements. Understanding the ignorance of a measure is crucial. In addition, the values computed by the measures may not all be useful. For instance, most terminological measures return a high similarity value between *mouse* and *mouse*, but these elements may not be related (one if a reference to the animal and the other to a computer device). In such case, a contextual similarity measure may disambiguate the two words, and the value computed by this contextual measure is discriminative for the pair of elements. Thus, it is important to measure the ignorance of a similarity measure. For those reasons, the selection and the tuning of the similarity measures for selecting the correspondences are one of the ten challenges identified by Shvaiko and Euzenat (Shvaiko and Euzenat, 2008).

In this paper, we propose **a framework which aims at selecting the correspondences** independently of the similarity measures. It takes into account both the ignorance and the differences between the similarity measures and the discriminative similarity values of a candidate correspondence. In addition, our approach is flexible when one needs to add or remove a similarity measure. As a consequence, no tuning is required from the user to combine the similarity measures. The main contributions of this paper can be summarized as follows: (i) a flexible model for representing similarity measures and computing their dissimilarity and ignorance, (ii) a robust framework for selecting the correspondences regardless of the similarity measures and (iii) the validation of this approach by using a benchmark with real-world datasets. The rest of this paper is organized as follows: Section 2 presents the related work in data integration, and more specifically how matching approaches select the correspondences. Next, we provide in Section 3 the formal definitions for our problem. In Section 4, we describe our framework for selecting correspondences. An experimental validation is presented in Section 5. Finally, we conclude and outline future work in Section 6.

2 RELATED WORK

As the matching process covers different but related domains, these recent books provide the related work in **entity resolution** (Talburt, 2011), **schema matching** (Bellahsene et al., 2011) and **ontology alignment** (Euzenat and Shvaiko, 2007). Schema matching and ontology mainly deal with metadata (e.g., labels, properties, constraints) while entity matching is at the instance level. Yet, both use similarity measures to obtain similarity values between metadata or instances, although the types and nature of the similarity measures may differ. The **combination of similarity measures** and **the decision maker for selecting a pair as a correspondence** is a common issue in the matching domains.

To combine the results of different similarity measures, the simplest solution is to use a function (e.g., weighted average). For instance, most of the systems which participate in the **Ontology Alignment Evaluation Initiative** (Euzenat et al., 2011) compute initial similarity values with terminological, structural or instance-based measures, and they may refine the results by applying reasoning. The combination of the measures may be performed with more complex function, such as **artificial neural networks** (Gracia et al., 2011). This trend is confirmed in the entity matching task, where most tools combine the measures with a **numeric or rule-based function** (Köpcke and Rahm, 2010). Some approaches dynamically select the measures to be applied : **RiMOM** (Li et al., 2009) is a multiple strategy dynamic ontology matching system and it selects the best strategy (composed of one or several similarity measures) to apply according to the features of the ontologies to be matched. The schema matcher **YAM** also dynamically combines similarity measures according to machine learning techniques (Duchateau et al., 2009), with the benefit of automatically tuning the thresholds for each measure.

Most matching approaches are based on a threshold to select correspondences. In schema matching, **Glue** (Doan et al., 2003), **COMA++** (Aumueller et al., 2005), **Quickmig** (Drumm et al., 2007) or **ASID** (Bozovic and Vassalos, 2008) automatically select their correspondences given a threshold. The threshold value can be manually tuned, for instance with a Graphical User Interface as in **Agreement Maker** (Cruz et al., 2007). In a similar fashion, the entity matching approach **FEBRL** enables users to choose between a threshold or an automatic selection based on a nearest neighbor classifier (Christen, 2008). In (Panse et al., 2013), the authors use a probabilistic model to minimize the impact of correspondences which have been incorrectly selected. The ontology alignment system **S-Match++** does not return a global similarity value for a candidate pair of elements, but the type of relationship between the elements (e.g., subsomption) (Avesani et al., 2005).

Thus, the decision to select a pair as a correspondence mainly depends on the result of a SAT solver.

To summarize, our framework aims at tackling issues from the previously described systems. First, the similarity measures combined by a tool cannot have the same impact, because some of them are very similar for computing a score. On the contrary, the specificity of other measures should be reflected during the combination. Thus, our framework includes a model to classify these measures, and our matching approach is then independent from these measures, thus providing more **genericity**. To avoid the tuning task required by most matching tools, especially when adding or removing a similarity measure, our framework should be **flexible**: the combination of the similarity measures does not depend on the type of measure. A last remark deals with the similarity values: many values are not interesting enough to be exploited, and candidate correspondences do not obtain useful scores for all measures. Thus, our framework is **selective** because only the most interesting values should be used to compute a global similarity value for a candidate correspondence.

3 PRELIMINARIES

In this section, we formally define the problem and we present our running example.

3.1 Formalizing the Problem

The matching process deals with data sources, which can be schemas, ontologies, models and/or a set of instances. Let us consider a set of data sources $\mathcal{D}$. A data source $d \in \mathcal{D}$ is composed of a set of elements $\mathcal{E}_d$, each of them associated with an identifier and attributes. Optionally, these elements may be linked by relationships. The size of a data source, or its number of elements, is noted $|\mathcal{E}_d|$. The matching problem consists of discovering correspondences, i.e., links between the semantically-equivalent elements from different data sources. Let us note $\mathcal{S}$ the set of similarity measures used by a tool to discover these identical elements. To assess a degree of similarity between two elements $e \in \mathcal{E}_d$ and $e' \in \mathcal{E}_{d'}$, a similarity measure $sim \in \mathcal{S}$ computes a similarity value $sim(e,e')$ between these elements as follows:

$$sim(e,e') \rightarrow [0,1]$$

As explained in (Euzenat and Shvaiko, 2007), we assume that all similarity measures return values which can be normalized in the range $[0,1]$. Thus, this definition includes the similarity measures which return a value in $\Re$ or those which compute a semantic relationship (e.g., *equivalence* or *hyponymy*). Similarly to most matching approaches, we focus on one to one matching, i.e., a correspondence only involves two elements from different data sources.

Since a similarity measure mainly exploits a few properties of the data sources (e.g., an element attribute, the relationship between elements, etc.), the matching process often applies different similarity measures, thus producing a similarity matrix. All similarity values are then combined into a global similarity score. The combination function may be complex and require tuning. The set of final correspondences between d and d' is noted $\mathcal{M}(d,d')$ and it contains simple correspondences represented as a tuple $(e,\ e')$. These final correspondences are selected among all possible candidate correspondences (mainly the Cartesian product of $\mathcal{E}_d$ and $\mathcal{E}_{d'}$) according to their global similarity score (or confidence score). This selection is performed manually (e.g., by proposing a ranking of the correspondences with the highest global scores) or automatically (e.g., the matching process selects the correspondences with a global score above a given threshold). Note that the mapping function is not in the scope of this paper.

3.2 Running Example

Based on the previous definitions, we describe a running example. *For sake of clarity, it has been constrained to two data sources $d,d' \in \mathcal{D}$. Each of them contains three elements: $\mathcal{E}_d = \{a,b,c\}$ and $\mathcal{E}_{d'} = \{a',b',d'\}$. All possible pairs of elements are candidate correspondences, for which their similarity has to be verified. The set of correct correspondences (i.e., provided by an expert) contains two correspondences (a,a') and (b,b'). To match these two data sources, we use a set of four similarity measures $\mathcal{S} = \{sim_1, sim_2, sim_3, sim_4\}$. Table 1 presents the similarity matrices of each measure, i.e., the similarity values computed for each pair of elements. For instance, the pair of elements (a,a') obtains a similarity value equal to* 0.8 *with the measure sim_1*. Next, we illustrate our framework with this example.

4 A GENERIC FRAMEWORK TO SELECT CORRESPONDENCES

In this section, we describe our framework to select and combine the most relevant similarity values for a given pair of elements regardless of the number of similarity measures. The basic intuition is that

sim_1	a	b	c
a’	0.8	0	0
b’	0	0.3	0
d’	0.8	0	0.7

sim_2	a	b	c
a’	0.1	0.1	0.1
b’	0.2	0.1	0.2
d’	0.8	0.2	0.6

sim_3	a	b	c
a’	0.6	0.2	0.1
b’	0.3	0.9	0.4
d’	0.3	0.2	0.2

sim_4	a	b	c
a’	0	0	0.5
b’	0	0.5	0
d’	0	0	0

Table 1: Similarity Matrices for Similarity Measures sim_1, sim_2, sim_3, and sim_4 .

the confidence score of a pair (i.e., the global similarity value) has to reflect the presence of discriminative similarity values for that pair and the diversity of the types of similarity measures which computed these similarity values. In other words, a confidence score should be higher for a candidate correspondence which obtains discriminative similarity values with a terminological, a semantic and a constraint-based measures rather than for a candidate correspondence which obtains the same values with three terminological measures. In the rest of this section, we describe a model to classify similarity measures and compute their dissimilarity (Section 4.1). Then, we explain the meaning of a discriminative similarity value for a candidate correspondence (Section 4.2). The classification model and the discrimination definition can finally be used to compute a confidence score and select the correspondences (Section 4.3).

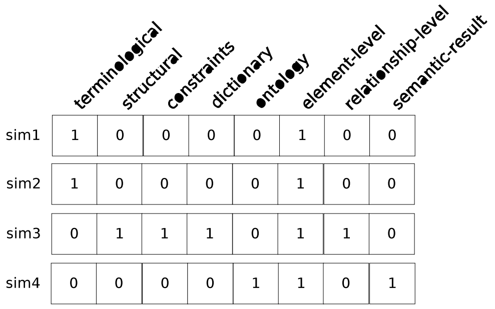
4.1 A Model for Comparing Similarity Measures

Matching approaches combine different types of similarity measures in order to exploit all properties of the data sources and to increase the chances of discovering correct correspondences (Euzenat and Shvaiko, 2007). However, one needs to correctly tune the matching tool when possible to avoid that a set of similar measures (e.g., terminological) has too much impact in the global similarity score. Indeed, the types and differences between similarity measures are rarely taken into account. In addition, their ignorance, or inability to detect a similarity, is not considered. For instance, two elements labeled *mouse* would be matched with a high score by a terminological measure, although one of them may refer to as a computer device while the other may stand for an animal. This is due to the fact that the terminological measure only uses the string labels to detect a similarity, and no other information such as external resource, constraints, element context, etc. Consequently, the ignorance of a similarity measure should reflect its limitations in terms of information that it uses to detect a similarity. Similarity measures have been largely studied in the literature (Euzenat and Shvaiko, 2007; Cohen et al., 2003). And a classification of these measures has been proposed (Euzenat et al., 2004) and later refined (Shvaiko and Euzenat, 2005). In a similar fashion, we provide a non-exhaustive list of features of similarity measures:

- the type or category (e.g., terminological, linguistic, structural)
- the type of input (e.g., character strings, records)
- the type of output (e.g., number, semantic relationship)
- the use of external resources (e.g., a dictionary, an ontology)

To **compare the similarity measures**, we propose to represent them as binary vectors according to their features. A feature is in our context a property that the similarity measure fulfills or not. Each similarity measure $sim_i \in \mathcal{S}$ is represented by a binary vector $v_i \in \mathcal{V}$. The size of a vector is $|v_i|$, i.e., the number of features. We can see the set of binary vectors as a matrix, where f_{ih} represents the binary value of the h^{th} feature for the i^{th} vector.

In our running example, the four similarity measures are represented by vectors with 8 *features, as shown in Figure 1. For instance, the third similarity measure is not terminological but it exploits the structure and the constraints of the data source with a dictionary as external resource. It is applied against the elements and the relationships of the data sources.*



	terminological	structural	constraints	dictionary	ontology	element-level	relationship-level	semantic-result
sim1	1	0	0	0	0	1	0	0
sim2	1	0	0	0	0	1	0	0
sim3	0	1	1	1	0	1	1	0
sim4	0	0	0	0	1	1	0	1

Figure 1: Binary Vectors for each Similarity Measure.

We finally obtain a classification of the similarity measures. This classification not only highlights the features of the measures but also indicates the ignorance of a measure (with respect to the features). For instance, a terminological measure such as sim_1 would not detect a similarity between synonyms in most cases. It is possible to refine the classification by adding more features. The goal is to compute the differences of each similarity measure with regards to the other ones. For each binary vector $v_i \in \mathcal{V}$, we compute its **difference score** noted Δ_{sim_i} with regards to the other vectors by applying Formula 1. The main intuition is that a vector is different from another one if its features are different. Thus, we analyse each feature of a vector and we calculate the rate of dissimilar values in the other vectors for the same feature. Given a number n of similarity measures and a number g of features, a difference score Δ_{sim_i} is computed as:

$$\Delta_{sim_i} = \frac{\sum_{h=1}^{g}(\sum_{j \neq i, j=1}^{n} \frac{f_{jh} \oplus f_{ih}}{n-1})}{g} \qquad (1)$$

The function $f_{jh} \oplus f_{ih}$ is the boolean operation *exclusive or*, which excludes the possibility of same values for both features. In other words, it returns 1 if the boolean features are different, 0 else. If all binary vectors are identical, the difference score equals 0, but this indicates that the vector representation of the measures is not detailed enough.

We then normalize this difference score in the range $[0,1]$ to obtain **the dissimilarity of a measure** with regards to others. This normalization is shown in Formula 2:

$$dissim_{sim_i} = \frac{\Delta_{sim_i}}{\sum_{a=1}^{n} \Delta_{sim_a}} \qquad (2)$$

The dissimilarity score measures the percentage of features which are different from other measures. We note that the following statement holds with the normalization: $\sum_{sim_i \in \mathcal{S}} dissim_{sim_i} = 1$.

Table 2 provides the difference and dissimilarity scores for each measure in our example. For instance, the difference score of the similarity measure sim_1 *equals* $\frac{\frac{2}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3}+\frac{1}{3}+0+\frac{1}{3}+\frac{1}{3}}{8} = 0.33$. *Its dissimilarity score is equal to* $\frac{0.33}{(0.33+0.33+0.67+0.375)} = 0.19$. *This means that the similarity measure* sim_1 *has* 19% *of different features compared to other measures, or* sim_1 *has an ignorance degree equal to* 81%.

	sim_1	sim_2	sim_3	sim_4
Δ	0.33	0.33	0.67	0.375
dissim	0.19	0.19	0.40	0.22

Table 2: Difference and Dissimilarity Scores of each Measure.

4.2 Discriminative Measures

In real-case scenarios, a correct correspondence would certainly not obtain a high similarity value for all measures. Therefore, combining the similarity values of all measures to compute a global score may not seem suitable. Besides, if a measure returns high similarity values for most candidate correspondences (e.g., a measure based on data types), then these high values may not be useful to disambiguate a conflict between two pairs of elements. Thus, we propose to discover which similarity measures are discriminative for a candidate correspondence, i.e., the measures which computed a significant value for that correspondence w.r.t. others.

To fulfill this goal, we are interested in evaluating the range inside which a value is considered as discriminative. We use **the mean and the standard deviation** to obtain this range of values. Jain et al. have demonstrated that these formulas are efficient when we do not need to estimate their values (Jain et al., 2005). Besides, the standard deviation is sensitive to extreme values, i.e., the ones that discriminate. Let us consider a similarity measure $sim_i \in \mathcal{S}$ which has computed a value for all possible correspondences, i.e., $|\mathcal{E}_d| \times |\mathcal{E}_{d'}|$. We can compute the average μ and the standard deviation σ of this measure sim_i as follows:

$$\forall e \in \mathcal{E}_d, \forall e' \in \mathcal{E}_{d'}, \quad \mu = \frac{\sum sim_i(e,e')}{|\mathcal{E}_d| \times |\mathcal{E}_{d'}|}$$

$$\sigma = \sqrt{\frac{\sum (sim_i(e,e') - \mu)^2}{|\mathcal{E}_d| \times |\mathcal{E}_{d'}|}}$$

As the standard deviation represents the dispersion of a value distribution, the range of values close to the average is given by:

$$rnd_{sim_i} = [\mu - \sigma, \mu + \sigma]$$

Note that the lower limit of that range equals to 0 if $\mu - \sigma < 0$, and the upper limit is equal to 1 if $\mu + \sigma > 1$. The similarity values in the range rnd_{sim_i} do not discriminate a candidate correspondence. Therefore, a discriminative similarity value for a candidate correspondence (e,e') should not belong to the range rnd_{sim_i}. We note $\gamma(e,e')$ **the set of measures that discriminate a candidate correspondence**, i.e., the measures that satisfy this condition: $\forall sim_i, sim_i \in \gamma(e,e') \iff sim_i(e,e') \notin rnd_{sim_i}$.

In our example, we can compute the average of the measure sim_1. *The nine values have an average and a standard deviation equal to* 0.28 *and* 0.35 *respectively. Consequently, the range of non-discriminative values is* $[0, 0.63]$. *For instance, the pair* (a,a') *is discriminated by the measure* sim_1 *since the value computed for this candidate correspondence* (0.8) *is not in that range. Note that all* <u>*underlined values*</u> *in the similarity matrices of Table 1 indicate that the corre-*

sponding measure is discriminative for the candidate correspondence. Thus, $\gamma(a,a') = \{sim_1, sim_3\}$.

By selecting a subset of the similarity measures, we express the fact that the discriminative similarity values have more impact than others during the correspondences selection. However, a candidate correspondence may not have any discriminative values in the first round, and thus it is not considered for computing its confidence score. To identify the discriminative measures for such candidate correspondences, we iterate the process by noting $\gamma^k(e,e')$ the k^{th} iteration of this process. At the end of each step, the previous similarity values are discarded, and we compute a new average and standard deviation, thus generating **a set of new discriminative measures** (or an empty set). This set is then merged into $\Gamma^t(e,e')$ as shown with this formula:

$$\Gamma^t(e,e') = \bigcup_{k=1}^{t} \gamma^k(e,e')$$

The question is how to determine a value for the number of iterations t. We can use the same techniques than the matching tools such as a threshold value (i.e., there is no more iteration if the average of similarity values reaches a threshold value) or the first k iterations. We can also iterate until all elements of a data source have been discriminated by at least one measure, and/or when an iteration has no more discriminative values to propose.

Let us compute the set of discriminative measures for the second iteration in our running example. We first discard the discriminative similarity values from the matrices that were previously identified (i.e., the underlined values). What happens for the similarity measure sim_1 *? The new average and standard deviation computed for the remaining* 6 *values equals* 0.04 *and* 0.24 *respectively. The similarity value* 0.3 *for* (b,b') *is a discriminative value at this iteration. Consequently,* sim_1 *is added in the set of discriminative measures for the candidate correspondence* (b,b') *and* $\Gamma^2(b,b') = \{sim_1, sim_3, sim_4\}$.

4.3 Computing a Confidence Score

The next step deals with the computation of **a confidence score** for a given candidate correspondence. This score is based on the discriminative similarity measures and their associated value. Indeed, the intuition is that we should be confident in a correspondence which obtains high similarity values computed by distinct and low-ignorance measures. Given a pair $m = (e,e')$ and its discriminative similarity values $< sim_1(e,e'), \ldots, sim_n(e,e') >$ with $sim_1, \ldots, sim_n \in \Gamma^t(e,e')$, the confidence score $conf^t(e,e')$ for the t^{th} iteration is calculated as follows:

$$conf^t_{(e,e')} = \sum_{i=1}^{n} dissim_{sim_i} \times \frac{\sum_{i=1}^{n} sim_i(e,e')}{n}$$

In this formula, we average the discriminative similarity values and we multiply the result by the sum of all dissimilarities of the measures. As both are in the range $[0,1]$, the confidence score also has values in the range $[0,1]$. Note that our formula is also a weighted average like in other approaches, however it does not require any tuning due to the independence and the model for the similarity measures.

Back to our example, we can compute the confidence scores of all candidate correspondences which have discriminative values at the first iteration. That is, the pairs (a, a'), (a, d'), (b, b'), (c, a'), and (c, d'). Let us detail the probability for the first pair to be correct:

$$conf(a,a') = (0.19 + 0.40) \times \frac{0.8 + 0.6}{2} = 0.41$$

The confidence score for the other candidate correspondences are: conf(a, d') = 0.30*, conf(b, b')* = 0.43*, conf(c, a')* = 0.19*, and conf(c, d')* = 0.25*. We notice that although the similarity values for the pairs (a, a') and (a, d') are close, we have more confidence in the former pair since it obtains discriminative similarity values with more dissimilar similarity measures. The candidate correspondence (c, a') is penalized by* sim_3 *since this measure mainly computes high similarity values, except for this candidate pair.*

4.4 Discussion

We finally discuss several points of our approach:

- When the confidence scores of two correspondences involving the same element are very close, they can be part of a complex correspondence. This needs to be checked by using refined techniques to discover these complex correspondences (Bilke and Naumann, 2005; Dhamankar et al., 2004; Saleem and Bellahsene, 2009).
- The boolean features are produced objectively. Designers or users of a similarity measure knows whether the measure satisfies a feature or not. A challenging perspective is the automatic definition of the binary vector. In the OAEI track *benchmark*[1], the objective is to detect the ability of a matching tool by duplicating a dataset with a minor change (e.g., changing the language, or deleting the annotations) (Euzenat et al., 2011). By

[1] Ontology Alignement Evaluation Initiative (January 2013), http://oaei.ontologymatching.org/

applying a similarity measure to this OAEI track, one is able to check if the measure is resistant to the change, and thus can compute the binary vector of that similarity measure.

- In the current version, we use binary vectors, which means that a measure owns the feature or not. We could relax this binary constraint by allowing real values. In that case, the vector shows the probability that the measure satisfies the feature, or the degree of ignorance for the feature. However, such a modification has an impact on the difference score formula.
- If we do not discriminate the similarity measures for a given candidate correspondence, the confidence score would be equal to the average of all similarity values computed for that candidate correspondence.
- It is possible that all similarity measures have the same similarity score, despite their different features. For instance, the four similarity measures of the running example could have obtained each a score equal to 0.25. In such case, the dissimilarity scores still have a significant impact, since they are used only for the discriminative values that have been calculated.
- Our approach can be plugged into most matching approaches, especially those that computes a global similarity score. Our confidence score may be used to select correspondences either manually (the user can select within the list of top-k correspondences based on their confidence score) or automatically (all confidence scores above a threshold are returned). Besides, our confidence score reflects the types and ignorance of the similarity measures which have computed the discriminative similarity values.
- Finally, our approach does not require any tuning for combining the similarity measures, because we select the values computed by similarity measures if they are sufficiently discriminative.

5 EVALUATION

In this section, we validate our approach in an Entity Matching context. We first describe the evaluation protocol (benchmark and another tool), then we compare our approach with another entity matching tool in terms of matching quality, and we finally show the robustness of our approach when adding new similarity measures.

5.1 Evaluation Protocol

To demonstrate the benefits of our framework, we have implemented it and tested against a benchmark for entity resolution (Kopcke et al., 2010). This benchmark[2] contains four datasets and has been evaluated by their authors with a matching tool (whose name is unknown due to licensing). We refer to this matching tool as *BenchTool* in the rest of the paper. The four datasets mainly cover two domains: Web products (*Abt-Buy* and *Amazon-GoogleProducts*) and publications (*DBLP-Scholar* and *DBLP-ACM*). The size of the data sources contained in these datasets vary from 1081 (*Abt*) entities to 64283 (*Scholar*), with an average around 2000 entities. The set of perfect correspondences is provided for each dataset and their size varies from 1097 (*Abt-Buy*) to 5347 (*DBLP-Scholar*) (Kopcke et al., 2010). The matching quality is computed with the three well-known metrics (Euzenat and Shvaiko, 2007; Bellahsene et al., 2011). Precision calculates the proportion of relevant correspondences among the discovered ones. Recall computes the proportion of correct discovered correspondences among all correct ones. Finally, the F-measure evaluates the harmonic mean between precision and recall. Since both tools obtain a score for these metrics which is close to the F-measure (e.g., precision equal to 97%, recall to 95% and F-measure at 96%), we only present the plots for the F-measure.

The configuration of our framework is as follows. We have 5 similarity measures : three of them are terminological (Jaro Winkler, Monge Elkan, Smith Waterman from the Second String API[3]), another one is based on the frequency of the words in fields such as description or title (Duchateau et al., 2009), and the last one is the Resnik measure applied to the Wordnet dictionary (Resnik, 1999). All these measures have been classified with the 8 features described in Section 4.1. The number of iterations is limited to 2 and the conditions for selecting a correspondence is a combination of threshold and top-1: for each element from the source, its correspondence is the candidate correspondence with the best confidence score only if this score is above a threshold. This threshold is computed by averaging all similarity values. For the *BenchTool*, we assume that the best tuning was performed by the authors when they ran it against the Entity Matching Benchmark (Kopcke et al., 2010).

[2] Entity Matching benchmark (January 2013), `http://dbs.uni-leipzig.de/en/research/projects/object_matching/`

[3] API at `http://secondstring.sourceforge.net/`

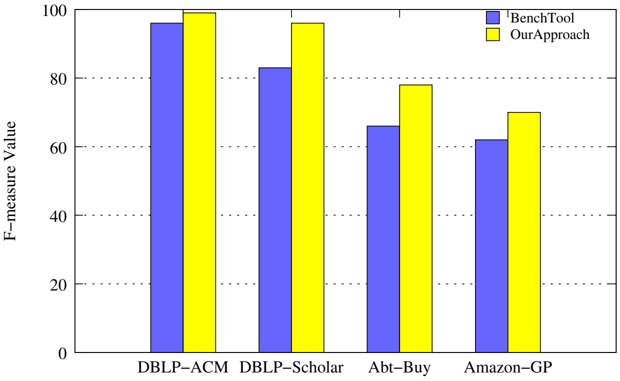


Figure 2: Results of BenchTool and our approach in terms of F-measure for the 4 datasets.

5.2 A Quality Comparison with BenchTool

Our first experiment aims at showing the matching quality obtained by our approach with regards to the *BenchTool* approach. Figure 2 depicts the results of the two systems for the four datasets. For the *publications* datasets, which are easier to match, both tools obtain an acceptable matching quality and our approach achieves a F-measure score above 95%. The *products* datasets are more difficult to match for two reasons: first, a *description* field is confusing because it contains either full sentences or sets of keywords. Secondly, two products may slightly differ (e.g, two hard drives of different size from the same manufacturer). For these datasets, *BenchTool* obtains a F-measure around $60 - 65\%$. Our approach performs better with a F-measure between 70% to 78%. Although our tool was not specifically designed for the entity matching task, we note that it achieves a better F-measure for all datasets. This means that our generic framework for selecting correspondences independently from the similarity measures is effective.

5.3 Demonstrating Robustness and Flexibility

In a second experiment, we demonstrate the robustness and the flexibility of our approach regardless of the similarity measures. A majority of matching approaches requires some tuning for combining similarity measures (e.g., setting weights). Thus, when a new similarity measure is added, it is necessary to reconfigure the tool. Since our framework considers each similarity measure individually, there is no need for tuning[4] and we demonstrate that the matching quality does not decrease when adding more measures. We have compared the variation of the F-measure value when increasing the number of similarity measures. To perform this experiment, five measures from the Second String API have been added (e.g., Affine Gap, Jaccard), and the measures have been randomly selected. We have run this experiment 10 times for each dataset to limit the impact of the randomness. Figure 3 illustrates the average F-measure (of all runs) for the four datasets with various number of similarity measures. When adding more measures and without any tuning, the trend of the plots is an increase or stabilization of the F-measure value. For the *DBLP-Scholar* dataset, the F-measure value is equal to 70% with one measure, and then it increases to reach 98% with 10 measures. For the *products* datasets, the 5 terminological measures which have been added are not sufficient to improve strongly the results. Other types of similarity measures are necessary to increase the matching quality of these datasets. To conclude, our framework is robust and flexible to the number of similarity measures that can be added without tuning.

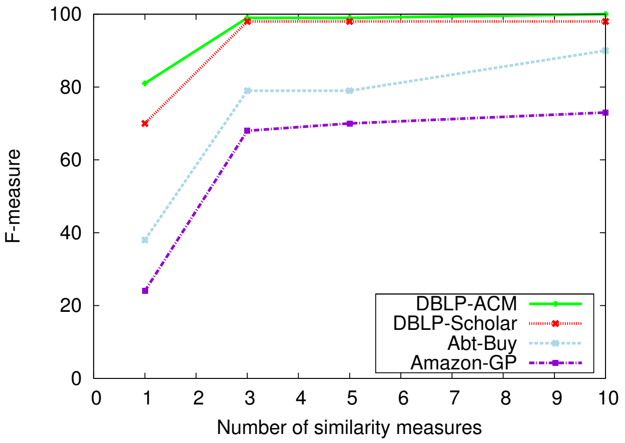


Figure 3: Results of our approach for the 4 datasets when varying the number of similarity measures.

6 CONCLUSION

In this paper, we have proposed a novel framework for selecting correspondences in a matching or ontology problem. Contrary to other approaches, our approach does not require any tuning to combine those

[4] The similarity measure has to be described according to the boolean features, which is still simpler than tuning a combination function. And such description can be shared and reused in an ontology for instance.

measures. The experiments against an entity resolution benchmark have demonstrated both the major improvement of our approach in terms of quality and its robustness regardless of the number of similarity measures involved in the matching. As for the perspectives, we plan to perform more experiments, both with other data integration tasks (schema matching, ontology alignment) and with various configurations of the parameters (number of iterations, number of features). Although the definition of features for the similarity measures can be designed in an ontology and shared with others users, it would be interesting to automatically compute the dissimilarity scores, for instance by analyzing the distribution values of the measures. Converting the binary vectors into real-valued vectors would refine the degree of ignorance of the measures. Such vectors may be computed with specific datasets, in which a minor change reflects a feature.

REFERENCES

Aumueller, D., Do, H. H., Massmann, S., and Rahm, E. (2005). Schema and ontology matching with COMA++. In *ACM SIGMOD*, pages 906–908.

Avesani, P., Giunchiglia, F., and Yatskevich, M. (2005). A large scale taxonomy mapping evaluation. In *International Semantic Web Conference*, pages 67–81.

Bellahsene, Z., Bonifati, A., and Rahm, E. (2011). *Schema Matching and Mapping*. Springer-Verlag, Heidelberg.

Bernstein, P. A., Madhavan, J., and Rahm, E. (2011). Generic schema matching, ten years later. *PVLDB*, 4(11):695–701.

Bilke, A. and Naumann, F. (2005). Schema matching using duplicates. *ICDE*, 0:69–80.

Bozovic, N. and Vassalos, V. (2008). Two-phase schema matching in real world relational databases. In *ICDE Workshops*, pages 290–296.

Christen, P. (2008). Febrl -: an open source data cleaning, deduplication and record linkage system with a graphical user interface. In *SIGKDD International Conference on Knowledge Discovery and Datamining*, KDD'08, pages 1065–1068. ACM.

Cohen, W., Ravikumar, P., and Fienberg, S. (2003). A comparison of string distance metrics for name-matching tasks. In *In Proceedings of the IJCAI-2003*.

Cruz, I. F., Sunna, W., Makar, N., and Bathala, S. (2007). A visual tool for ontology alignment to enable geospatial interoperability. *J. Vis. Lang. Comput.*, 18(3):230–254.

Dhamankar, R., Lee, Y., Doan, A., Halevy, A., and Domingos, P. (2004). iMAP: Discovering Complex Semantic Matches between Database Schemas. In *ACM SIGMOD*, pages 383–394.

Doan, A., Madhavan, J., Dhamankar, R., Domingos, P., and Halevy, A. Y. (2003). Learning to match ontologies on the semantic web. *VLDB J.*, 12(4):303–319.

Drumm, C., Schmitt, M., Do, H. H., and Rahm, E. (2007). Quickmig: automatic schema matching for data migration projects. In *CIKM*, pages 107–116. ACM.

Duchateau, F., Coletta, R., Bellahsene, Z., and Miller, R. J. (2009). (Not) Yet Another Matcher. In *CIKM*, pages 1537–1540.

Euzenat, J. et al. (2004). State of the art on ontology matching. Technical Report KWEB/2004/D2.2.3/v1.2, Knowledge Web.

Euzenat, J., Ferrara, A., van Hage, W. R., Hollink, L., Meilicke, C., Nikolov, A., Ritze, D., Scharffe, F., Shvaiko, P., Stuckenschmidt, H., Sváb-Zamazal, O., and dos Santos, C. T. (2011). Results of the ontology alignment evaluation initiative 2011. In *OM*.

Euzenat, J. and Shvaiko, P. (2007). *Ontology matching*. Springer-Verlag, Heidelberg (DE).

Fellegi, I. P. and Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association*, 64:1183–1210.

Gracia, J., Bernad, J., and Mena, E. (2011). Ontology matching with cider: evaluation report for oaei 2011. In *OM*.

Jain, A., Nandakumar, K., and Ross, A. (2005). Score normalization in multimodal biometric systems. *Pattern Recognition*, 38(12):2270–2285.

Köpcke, H. and Rahm, E. (2010). Frameworks for entity matching: A comparison. *Data Knowl. Eng.*, 69:197–210.

Kopcke, H., Thor, A., and Rahm, E. (2010). Learning-based approaches for matching web data entities. *IEEE Internet Computing*, 14(4):23–31.

Li, J., Tang, J., Li, Y., and Luo, Q. (2009). Rimom: A dynamic multistrategy ontology alignment framework. *IEEE Trans. on Knowl. and Data Eng.*, 21(8):1218–1232.

Panse, F., Ritter, N., and van Keulen, M. (2013). Indeterministic handling of uncertain decisions in deduplication. *Journal of Data and Information Quality*.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130.

Saleem, K. and Bellahsene, Z. (2009). Complex schema match discovery and validation through collaboration. In *OTM Conferences (1)*, pages 406–413.

Shvaiko, P. and Euzenat, J. (2005). A survey of schema-based matching approaches. *Journal of Data Semantics IV*, pages 146–171.

Shvaiko, P. and Euzenat, J. (2008). Ten challenges for ontology matching. In *OTM Conferences (2)*, pages 1164–1182.

Talburt, J. R. (2011). *Entity Resolution and Information Quality*. Elsevier.

Winkler, W. E. (2006). Overview of record linkage and current research directions. Technical report, Bureau of the Census.