Non-Binary Evaluation for Schema Matching

Tomer Sagi, Avigdor Gal

Technion – Israel Institute of Technology, Haifa, Israel

Abstract. In this work we extend the commonly used binary evaluation of schema matching to support evaluation methods for non-binary matching results as well. We motivate our work with some new applications of schema matching. Non-binary evaluation is formally defined together with a set of three new, non-binary evaluation measures using a vector-space representation of schema matching outcome. We provide an empirical evaluation to support the usefulness of non-binary evaluation and show its superiority to its binary counterpart.

1 Introduction

Schema matching is the task of providing correspondences between concepts describing the meaning of data in various heterogeneous, distributed data sources (*e.g.* attributes in database schemata and tags in XML DTDs). In its origin, schema matching was conceived to be a preliminary process to schema mapping. A basic assumption that accompanied this research field from its inception is that schema matching provides a set of definite (true or false) correspondences to be then validated by some human expert before mapping expressions are generated. The evaluation scheme for schema matching follows this assumption closely, making an extensive use of *Precision* and *Recall* [14] that were borrowed from the field of Information Retrieval. These measures provide a common-sense interpretation to our intuition of what is a "correct" matching by comparing a selected set of attribute correspondences against a set of attribute correspondences that is compiled by some domain expert.

Over the years, schema matching research has expanded and specialized to answer research and application questions in a variety of domains. Recent research is shifting focus from identifying new matching algorithms to methods for using the various matchers to efficiently and effectively solve a specific problem at hand. Thus, recent work focuses on selecting appropriate matchers [7], evaluating matchers with respect to a specific schema pair at hand [20,15], tuning matcher parameters [13,15], and ensembling results from different matchers [1,12]. These changes create a major focus shift for schema matching evaluation as well: from providing a final judgement of quality at the end of the matching process to a tool for intermediate assessment in the construction and application of matching tasks.

Existing measures no longer suffice when the need arises to evaluate non-binary intermediate results, with limited (and possibly non-existing) expert input. In particular, since existing evaluation measures are defined over binary

results, system designers are forced to evaluate the final outcome with no insight towards the impact of intermediate results on its performance.

In this work we propose new evaluation measures for *non-binary* schema matching outcomes. We devise these measures using an extension of the existing similarity matrix representation [9] onto a finite real vector space. Two of the measures extend, in a natural manner, the traditional Precision and Recall measures while the third measure provides a new perspective, called *Drift*, which measures the ability of a matcher to improve on the performance of other matchers. We have experimented with the proposed measures and report on their performance using real-world data. We thereby make the following contributions:

- At the conceptual level we introduce a framework for assessing the quality of a non-binary matching result.
- We propose three new non-binary evaluation measures for schema matching, formally analysing their properties.
- We provide an empirical evaluation, showing the benefits of using non-binary measures in common schema matching tasks.

The rest of the paper is organized as follows. Background on schema matching (Section 2) is followed by a schema matching evaluation model (Section 3). We introduce new measures in Section 4, followed by empirical evaluation (Section 5). We conclude with related work (Section 6) and summary (Section 7).

2 Preliminaries

The following schema matching model is based on [9]. Let schema $S = \{a_1, a_2, \ldots, a_n\}$ be a finite set of attributes. Attributes can be both simple and compound, compound attributes should not necessarily be disjoint, *etc.* For any schema pair $\{S, S'\}$, let $\mathcal{S} = S \times S'$ be the set of all possible attribute correspondences between S and S'.

Let $M(S, S')$ be an $n \times n'$ *similarity matrix* over $\mathcal{S}$, where $M_{i,j}$ (typically a real number in $[0, 1]$) represents a degree of similarity between the i-th attribute of S and the j-th attribute of S'. $M(S, S')$ is a *binary* similarity matrix if for all $1 \leq i \leq n$ and $1 \leq j \leq n_0$, $M_{i,j} \in \{0, 1\}$. A (possibly binary) similarity matrix is the output of the matching process. For any schema pair (S, S'), let the power-set $\Sigma = 2^{\mathcal{S}}$ be the set of all possible *schema matches* between this pair of schemata.

Example 1. Table 1 presents two similarity matrices between two simplified schemata, with four and three attributes, respectively. The similarity matrix on the left is the outcome of a matching process, using some matcher. The similarity matrix on the right is a binary similarity matrix, generated as the outcome of a decision-maker matcher [9] . This matcher enforces a binary decision while requiring participation of each attribute in at most one correspondence. □

$S_1 \longrightarrow$ / $\downarrow S_2$	1 cardNum	2 city	3 arrivalDay	4 checkIn Time
1 clientNum	0.84	0.32	0.32	0.30
2 city	0.29	1.00	0.33	0.30
3 checkInDate	0.34	0.33	0.35	0.64

$S_1 \longrightarrow$ / $\downarrow S_2$	1 cardNum	2 city	3 arrivalDay	4 checkIn Time
1 clientNum	1	0	0	0
2 city	0	1	0	0
3 checkInDate	0	0	0	1

Table 1: A Similarity Matrix Example

Matching schemas is often a stepped process in which different algorithms, rules, and constraints are applied. Several classifications of schema matching steps have been proposed over the years (see *e.g.*, [13,4]). Following Gal and Sagi [12], we separate matchers into those that are applied directly to the problem (*first-line matchers (1LM)*) and those that are applied to the outcome of other matchers (*second-line matchers (2LM)*). 1LMs receive two schemata and return a similarity matrix. 2LMs, which are often decision makers, receive a similarity matrix and return a (usually binary) similarity matrix.

In this work we tackle the limitation of current evaluation measures, restricted to evaluating binary 2LM results.

3 Schema Matching Evaluation

In this section we formally define similarity spaces (Section 3.1) and schema matching evaluation (Section 3.2).

3.1 Similarity Spaces

We propose a vector space representation of schema matching outcome to support the evaluation task. The notion of a similarity space is adopted from Zobel and Moffat [21] in the context of document vector spaces, where cosine similarity was used as a measure of similarity between document vectors. For convenience, we maintain matrix notation when referring to a dimension, marking a dimension as an (i, j) coordinate.

Definition 1. *Given schemata S and S', a* similarity space $\mathcal{V}_{\mathcal{S}}(S, S') = [0,1]^{|\mathcal{S}|}$ *is an $|\mathcal{S}|$-dimension vector space such that each dimension (i, j) in $\mathcal{V}_{\mathcal{S}}(S, S')$ corresponds to the attribute pair (a_i, a_j) in $\mathcal{S}$.*
For each dimension (i, j) in $\mathcal{V}_{\mathcal{S}}(S, S')$ let $v^{i,j} = (0, 0, ..., 1, ..., 0)$ be a vector with all 0 values except the (i, j) element, assigned with a 1 value.
Given a similarity matrix M over $\mathcal{S} = S \times S'$ and a similarity space $\mathcal{V}_{\mathcal{S}}(S, S')$, a similarity vector $\mathbf{v}(M)$ *from the space $\mathcal{S}$ is the vector:*

$$\mathbf{v}(M) = \sum_{(i,j)\ dimension\ in\ \mathcal{V}_{\mathcal{S}}(S,S')} M_{i,j} v^{i,j} \tag{1}$$

Whenever the referenced schemata S and S' and the similarity matrix M are clear from the context we use $\mathcal{V}$ as a shorthand notation of $\mathcal{V}_{\mathcal{S}}(S, S')$ and $\mathbf{v}$ as a shorthand notation of $\mathbf{v}(M)$.

	3 arrivalDay	4 CheckInTime
3 checkInDay	0.35	0.64

(a) Similarity Matrix

	3 arrivalDay	4 CheckInTime
3 checkInDay	0	1

(b) Binary Similarity Matrix

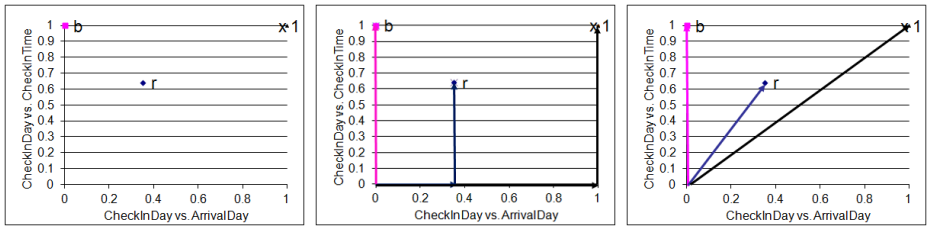
	3 arrivalDay	4 CheckInTime
3 checkInDay	1	1

(c) Exact Match Matrix

Table 2: Partial Similarity Matrix Examples

Each entry in a matrix M over $\mathcal{S}$ is represented as a similarity vector in $\mathcal{V}$. Therefore, a similarity vector represents the similarity of **pairs** of attributes.

Example 2. Consider Example 1. Let $\mathcal{V}$ be the vector space representation of the schema pair in Table 1. We present a simplified space defined over a single attribute from S_1, {checkInDay} and two attributes from S_2, {arrivalDay, CheckInTime}. Tables 2(a) and 2(b) show the relevant part of the similarity matrices in Table 1, respectively. Table 2(c) illustrates the relevant part of an exact match for the matching of checkInDay with arrivalDay and CheckInTime, as a binary similarity matrix. It is worth noting that while the decision maker chose to match checkInDay only with arrivalDay, the exact match matches checkInDay with both arrivalDay and CheckInTime. We can now visualize the 2-dimensional similarity vectors of the three matrices in Figure 1a. Each similarity vector is represented as a single point where vectors **r, b**, and **x** represent the matrices in tables 2(a), 2(b) and 2(c) respectively. □



(a) (b) (c)

Fig. 1: 2-Dimensional example

Example 2 demonstrates the basic elements of similarity spaces. Using a similarity space, we can measure the distance between two vectors by using standard distance functions. To do so, we use norms, functions that assign a strictly positive length or size to all vectors in a vector space, other than the

zero vector. We now present a formal definition of a similarity norm, adapted from the general definition of norms [3].

Definition 2. *Given a similarity space $\mathcal{V}_\mathcal{S}$ over $\mathcal{S}$, a* similarity norm *(or simply a* norm*) on $\mathcal{V}_\mathcal{S}$ is a function $p : \mathcal{V}_\mathcal{S} \rightarrow [0, 1]$ with the following properties:*

1. $\forall a \in [0, 1] \wedge \forall \mathbf{u}, \mathbf{v} \in \mathcal{V}_\mathcal{S}\,;\, p(a\mathbf{v}) = |a|p(\mathbf{v})$ *(positive scalability).*
2. $p(\mathbf{u} + \mathbf{v}) \leq p(\mathbf{u}) + p(\mathbf{v})$ *(subadditivity).*
3. *If $p(\mathbf{v}) = 0$ then $\mathbf{v}$ is the zero vector.*

A consequence of positive scalability and subadditivity is that $p(\mathbf{0}) = 0$ and $p(\mathbf{v}) \geq 0$, known as the positivity property. Every norm gives rise to a distance measure by defining: $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$. Well known examples are:

1. Manhattan distance, based upon the Manhattan norm (L_1) where the distance between any two vectors, is the sum of the differences between corresponding coordinates: $d_M(x, y) = \|x - y\|_1 = \sum_{i=1}^{n} |x_i - y_i|$
2. Euclidean Norm (L_2) gives rise to the Euclidean distance:
$$d_E(x, y) = \|x - y\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$
3. Maximum norm, gives rise to the Chebyshev distance: $d_{Chebyshev}(x, y) = \|x - y\|_\infty = \max_i (x_i - y_i)$

The decision of a specific norm to determine vector lengths allows us to define distance measures between vectors. The difference in norms impacts the distance computation as shown in the following example.

Example 3. Returning to our running example, figures 1c and 1b illustrate the length of vectors **b**,**r** and **x** using the Manhattan ($L1$) and Euclidean ($L2$) norms respectively. Comparing the figures, it is apparent that the choice of norm impacts the relative length of vectors. Using L_1 the lengths of **b** ,**r** and **x** are 1, 1, and 2 respectively. Using L_2 results in the lengths of **b** ,**r** and **x** being 1, 0.72, and 1.41 respectively. As demonstrated by this simple example, when using L_1, the lengths of **r** (the result vector) and **b** (the binary vector generated from it using a constraint function) are the same while when using L_2 their lengths differ. □

3.2 Schema Matching Evaluation

Schema matching evaluation is a task aiming at assessing the quality of a matching result.

Definition 3. *Let $\mathcal{M} = \{M_1, M_2, ..., M_n\}$ be a set of similarity matrices over $\mathcal{S} = S \times S'$. A* schema matching evaluation method *is a function*

$$g : \mathcal{M} \times \mathcal{M} \times ... \times \mathcal{M} \rightarrow \mathbb{R}$$

g receives a set of similarity matrices representing schema matching results and evaluates them to return a single real value. We can designate one of the similarity matrices to be an exact match, representing some "correct" match, typically provided by an expert. Such an exact match can be encoded as a binary similarity matrix, where "correct" correspondences are assigned the value of 1 and incorrect correspondences are assigned a value of 0.

Whenever an exact match is part of the input to g, the evaluation is performed with respect to it. For example, let $\mathcal{M} = \{M, M^e\}$ be a pair of two similarity matrices over $\mathcal{S} = S \times S'$. M is a binary matrix, representing the outcome of a decision maker schema matcher. M^e is a binary matrix, representing the exact match. Let $g^{BR}(M, M^e)$ be the well-known Recall evaluation method, computed as follows:

$$g^{BR}(M, M^e) = \frac{\mathbf{v}(M) \cdot \mathbf{v}(M^e)}{\|\mathbf{v}(M^e)\|_1} \tag{2}$$

where $\mathbf{v}(\cdot)$ is a binary similarity vector over $\mathcal{V}_{\mathcal{S}}(S, S')$ and $\|\cdot\|_1$ represents the Manhatten (L_1) norm.

4 Non-Binary Evaluation Measures

Evaluators evaluate similarity between vectors that are not necessarily binary. In this section we introduce three new, non-binary evaluators.

The first two non-binary evaluators extend common schema matching evaluators, namely Precision and Recall, to support non-binary values as well. A natural extension is achieved by simply removing the requirement of $\mathbf{v}(M)$ to be a binary vector. This simple extension enables evaluation of non-binary similarity vectors generated by 1LMs without applying any match selection rule or constraint and thus allowing for independent evaluation of 1LMs. We define these measures as follows:

Definition 4 (Non-binary Precision and Recall). *Let $\mathcal{V}_{\mathcal{S}}(S, S')$ be a similarity space and let $\mathbf{v}(M^e)$ be a similarity vector over $\mathcal{V}_{\mathcal{S}}$, where M^e represents an exact match.*
NBPrecision *is defined to be*

$$g^{NBP}(M, M^e) = \frac{\mathbf{v}(M) \cdot \mathbf{v}(M^e)}{\|\mathbf{v}(M)\|_1} \tag{3}$$

NBRecall *is defined to be*

$$g^{NBR}(M, M^e) = \frac{\mathbf{v}(M) \cdot \mathbf{v}(M^e)}{\|\mathbf{v}(M^e)\|_1} \tag{4}$$

*In both cases, $\mathbf{v}(M)$ is a **non** binary similarity vector over $\mathcal{V}$.*

Note that Eq. 2 and Eq. 4 differ in the type of their input vector $\mathbf{v}(M)$. A further relaxation of the evaluation measure can be done by removing the requirement that $\mathbf{v}(M^e)$ be a *binary* similarity vector. Such relaxation can support probabilistic schema matching [5,10], where probabilistically correct correspondences are assigned non-0 values in the similarity matrix and the matrix value as a whole represents a probability space over the set of attribute correspondences. We defer a discussion of this relaxation to future work.

To investigate the relationship between the binary and non-binary variations we consider next a special type of 2LMs we term *filters*. Such matchers decide for each entry of the similarity matrix whether its value should remain unchanged or reduced to 0. Filters are the non-binary equivalent to decision maker 2LMs in the binary world. Formally,

Definition 5 (Filters). *Let* $\mathbf{v}(M)$ *be a similarity vector, and let* $\mathbf{v}\left(M^b\right)$ *be a similarity vector returned by a 2LM.* $\mathbf{v}\left(M^b\right)$ *is* filtered *if*

$$\mathbf{v}\left(M^b\right)_{i,j} = \begin{cases} \mathbf{v}(M)_{i,j} & \textit{decision: unchanged} \\ 0 & \textit{decision: filtered} \end{cases}$$

The following proposition shows that NBRecall behavior in the non-binary setting is similar to that of Recall in its binary counterpart. For both, removing correspondences (by setting their similarity to 0) cannot improve performance.

Proposition 6. *Let* M *be some similarity matrix resulting from the application of some 1LM and let* M^b *be the result of a filter 2LM. Let* M^e *be an exact match. Then, for* $\mathbf{v}(M)$ *and* $\mathbf{v}(M^e)$ *the following holds:*

$$g^{NBR}(M, M^e) \geq g^{NBR}\left(M^b, M^e\right)$$

Proof. $g^{NBR}(M, M^e) = \frac{\mathbf{v}(M)\cdot\mathbf{v}(M^e)}{\|\mathbf{v}(M^e)\|_1}$ by Definition 4 $g^{NBR}\left(M^b, M^e\right) = \frac{\mathbf{v}\left(M^b\right)\cdot\mathbf{v}(M^e)}{\|\mathbf{v}(M^e)\|_1}$ by Definition 4 By way of contradiction, assume that

$$\mathbf{v}\left(M^b\right)\cdot\mathbf{v}(M^e) > \mathbf{v}(M)\cdot\mathbf{v}(M^e)$$

$$\Rightarrow \exists(i,j) \mid \mathbf{v}(M^e)_{i,j} = 1 \wedge \mathbf{v}(M)_{i,j} = 0 \wedge \mathbf{v}\left(M^b\right)_{i,j} > 0$$

which contradicts the assumption that $\mathbf{v}\left(M^b\right)$ is zero biased. Therefore,

$$\mathbf{v}\left(M^b\right)\cdot\mathbf{v}(M^e) \leq \mathbf{v}(M)\cdot\mathbf{v}(M^e)$$

$$\Rightarrow \frac{\mathbf{v}\left(M^b\right)\cdot\mathbf{v}(M^e)}{\|\mathbf{v}(M^e)\|_1} \leq \frac{\mathbf{v}(M)\cdot\mathbf{v}(M^e)}{\|\mathbf{v}(M^e)\|_1}$$

since $\|\mathbf{x}\| > 1$

Our third evaluator is designed to give further insight into the effects of applying various 2LM on 1LM results. NBPrecision and NBRecall evaluate the performance of 1LM irrespective of the application of subsequent 2LM. Classic Precision and Recall evaluate the combined effect of 1LM and 2LM. However, we cannot differentiate the impact each one has on the final outcome. To this effect we now present a new measure, which we name *Drift*.

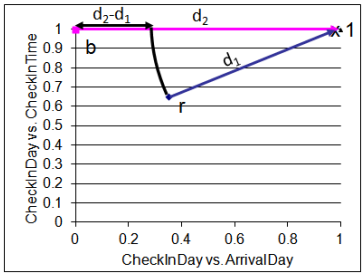


Fig. 2: Illustration of Drift based on Euclidean distance

Definition 7 (Drift). *Let* $\mathcal{V}_{\mathcal{S}}(S, S')$ *be a similarity space and let* $\mathbf{v}(M^e)$ *be a similarity vector over* $\mathcal{V}_{\mathcal{S}}$*, where* M^e *represents an exact match.*
Let $\mathbf{v}(M_1)$ *and* $\mathbf{v}(M_2)$ *be similarity vectors such that* $\mathbf{v}(M_1)$ *is the result of some 1LM and* $\mathbf{v}(M_2)$ *is the result of the application of some 2LM on* $\mathbf{v}(M_1)$*. Let* d_p *be a distance measure defined over a similarity norm* p*.* Drift *is defined as:*

$$g^{DL_p}(\mathbf{v}(M_1), \mathbf{v}(M_2), \mathbf{v}(M^e)) = d_p(\mathbf{v}(M_2), \mathbf{v}(M^e)) - d_p(\mathbf{v}(M_1), \mathbf{v}(M^e))$$

For example, *Euclidean Drift* is defined as:

$$g^{DL_2}(\mathbf{v}(M_1), \mathbf{v}(M_2), \mathbf{v}(M^e)) = d_E(\mathbf{v}(M_2), \mathbf{v}(M^e)) - d_E(\mathbf{v}(M_1), \mathbf{v}(M^e))$$

Drift is designed to evaluate the effect a 2LM has on the results of a 1LM. Some 2LM may consistently cause the result to drift away from (towards) the correct result, *i.e.*, the distance between the 2LM result vector and the exact match vector is larger than the distance between the 1LM result and the exact match vector when following specific 1LMs. To facilitate this evaluation we require the results of evaluating the 1LM and the 2LM to be consistent with each other and therefore require a distance measure (for example Euclidean distance) based on a similarity norm that ensures subadditivity.

Returning to our running example. In Figure 2, g^{DL_2} is visualized by taking the exact match vector $\mathbf{x}$ as the center of a circle of radius $d_1 = \sqrt{(1-0.35)^2+(1-0.65)^2} = 0.545$ which is the distance between vector r and the exact match vector. The arc of this circle crosses the line denoting the distance between vector b and the exact match vector. The length of this line is given by $d_2 = \sqrt{(1-1)^2+(1-0)^2} = 1$. Drift in this case equals 0.455 and is positive, meaning the 2LM has taken us farther away from the exact match with respect to the previous distance achieved by the 1LM.

5 Empirical Evaluation

We empirically evaluated non-binary evaluation measures in two usage scenarios. Section 5.1 details the setup of our experiments. We then evaluate the behavior of non-binary vs. binary Precision and Recall in Section 5.2 and evaluate the usage of L2 distance measures to determine 2LM drift in Section 5.3.

5.1 Setup

Evaluations were conducted on an Intel(R) Core(TM)2 Quad CPU Q8200 @ 2.33GHz. An experiment system was coded in Java, JDK version 1.6.0. JVM was initiated with 1.00GB heap-memory.

We use the *webform* dataset[1] containing 249 ontologies, automatically extracted from Web forms using the OntoBuilder Extractor [11] and matched into pairs. For each pair, an exact match was defined manually.

We use matching algorithms from the OntoBuilder matching system. *OntoBuilder* is a research prototype, developed for matching ontologies from the deep Web. The following OntoBuilder 1LMs are used in the evaluation (for a detailed description, see [11,16]): Term, Value, Composition(Graph), Precedence, and Similarity Flooding (a re-implementation of the algorithm presented in [17]). The 2LM decision makers used in the evaluation where *maximum weighted bipartite graph (MWBG)*, *stable marriage (SM)* , and *Threshold(t)*, a simple threshold rule eliminating all correspondences with similarity measure less then t.

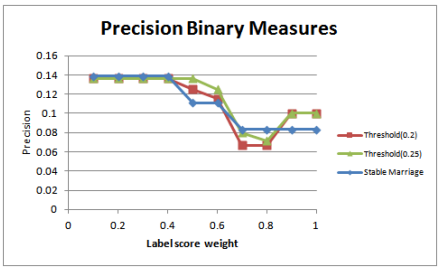
5.2 Using Non-Binary Measures for Tuning

To better understand the impact of using NBPrecision and NBRecal on the assessment of 1LMs we look into the tuning of a 1LM called *Term*, which calculates string similarity between fields of Web forms. Each score is based on two string elements, the field name (*name score*) and the field label (*label score*). The weight of each score is a tunable parameter. Current approaches, such as eTuner [13] use machine learning on test datasets to learn the optimal values for such parameters. Learning requires the matcher to return a binary similarity matrix and therefore, a 2LM of type decision maker is applied. In our case, we use Stable Marriage (SM) and *Threshold* to perform match selection.

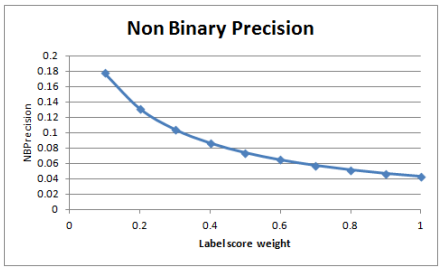
Figure 3 illustrates how binary and non-binary measures change in one schema pair when the weight of the label sub-matcher (X axis) is varied between 0 and 1. Figures 3a and 3c show how application of 2LMs SM and Threshold with 0.2 and 0.25 paint a different picture w.r.t. the effect of using label score than their non-binary counterparts (Figures 3b and 3d). The difference between line shapes is due to the fact that results of the 2LMs are evaluated using binary Precision and Recall and therefore jump between values as correspondences are either added or removed. Analyzing the variance reveals that differences between the results of the three 2LMs were due to one or two borderline correspondences that were added or removed by the arbitrary cut-off of threshold or the combined score of SM. Direct analysis of the 1LMs using non-binary measures, is insensitive to these small scale interactions and therefore provides a clear and consistent view of result quality.

An additional complication in matcher tuning is the requirement for an exact match, which is hard to come by. Generating an exact match between two schemata of considerable size is a daunting task. Furthermore, often the parameter being tuned may vary between datasets, requiring it to be tuned for each

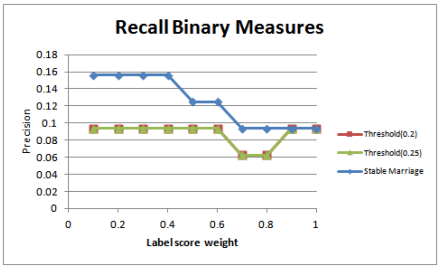
[1] https://bitbucket.org/tomers77/ontobuilder/wiki/Downloads



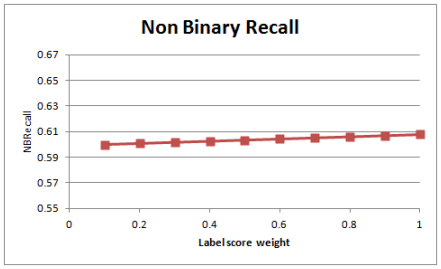
(a) Tuning by Precision



(b) Tuning by NBPrecision



(c) Tuning by Recall



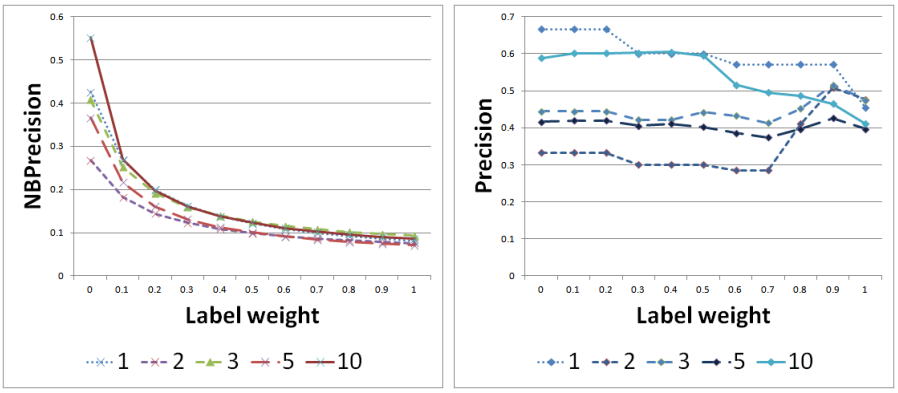
(d) Tuning by NBRecall

Fig. 3: Using various measures to tune *term algorithm*

dataset anew. These realities often cause designers to have very few exact pairs to work with. These designers would want the measure used to be robust to small sample sizes and its behavior to be stable even when only few pairs are available. In Figures 4a and 4b we present results of an experiment testing this stability. In the experiment we again varied the label score and compared Precision and NBPrecision over an increasing number of pairs. An outlier result in the second pair threw Precision completely off the trend and not until we reached 10 pairs did the Precision line converge back to the actual behavior. In contrast, NBPRecision maintained a stable line shape, varying in scale but not in form. These results demonstrate how binary measures mask the actual effect of using *label score* due to noise from small scale interactions between scores of different attribute correspondences. When compared to the smooth curve obtained from using NBPrecision and NBRecall, one immediately identifies the strength of these measures for tuning. Unsupervised learning as well as other automated tuning procedures benefit substantially from smooth curves creating a topology of performance which can be optimized.

5.3 2LM Drift

The following experiment aims at validating the performance of a 2LM, given an input of a 1LM. We hypothesize that some 2LMs consistently cause a drift



(a) (b)

Fig. 4: Impact of number of pairs on Precision vs. NBPrecision

from the correct result when following specific 1LMs. This drift if manifested as consistently increased distance between the result of the 2LM and the exact match vector $d_E\left(\mathbf{v}\left(M_2\right), \mathbf{v}\left(M^e\right)\right)$ than the distance between the 1LM result and the exact match vector $d_E\left(\mathbf{v}\left(M_1\right), \mathbf{v}\left(M^e\right)\right)$. We could then use this observation to correct the 2LM or search for one with a smaller drift.

We randomly chose 10 pairs from the web-form dataset and ran five 1LMs. The distance of these 1LM result vectors from the exact match vector is calculated and saved as d_1. We then run four 2LMs over the result vectors and save the distance of the 2LM result vectors from the exact match vector d_2. The difference between d_2 and d_1 represents the improvement (for negative values) or worsening (for positive values) caused by the 2LM over the 1LM result.

Results are presented in Figure 5 as box plots showing the distribution of Drift for different combinations of 1LMs and 2LMs. Stars and circles denote outlier schema pairs. Recall that negative values are more desirable since they indicate the 2LM has brought us closer to the exact match vector. From the 2LM perspective we observe that MWBG consistently reduces the distance from the exact match vector when coupled with all 1LMs except Similarity Flooding. Similar behavior is observed for Threshold and somewhat less consistently for Union. Dominants 2LM seems to present both inconsistent behavior and poor results, as it's distributions are wide and tend to be positive. SM presents mixed behavior, improving upon Precedence and Graph but causing an increase in distance when coupled with Value, Term and Similarity flooding. Examining the results from the 1LM perspective, Term achieves the best results when paired with Threshold and to a lesser extent MWBG. Value is best paired with Union, Threshold and MWBG. It seems that no 2LM is able to substantially improve the result of Similarity Flooding, this result indicates that the similarity matrices, generated by the 1LM are rather poor with little information for the 2LM to use. Precedence and Graph are easily improved by all 2LM, except Dominants,

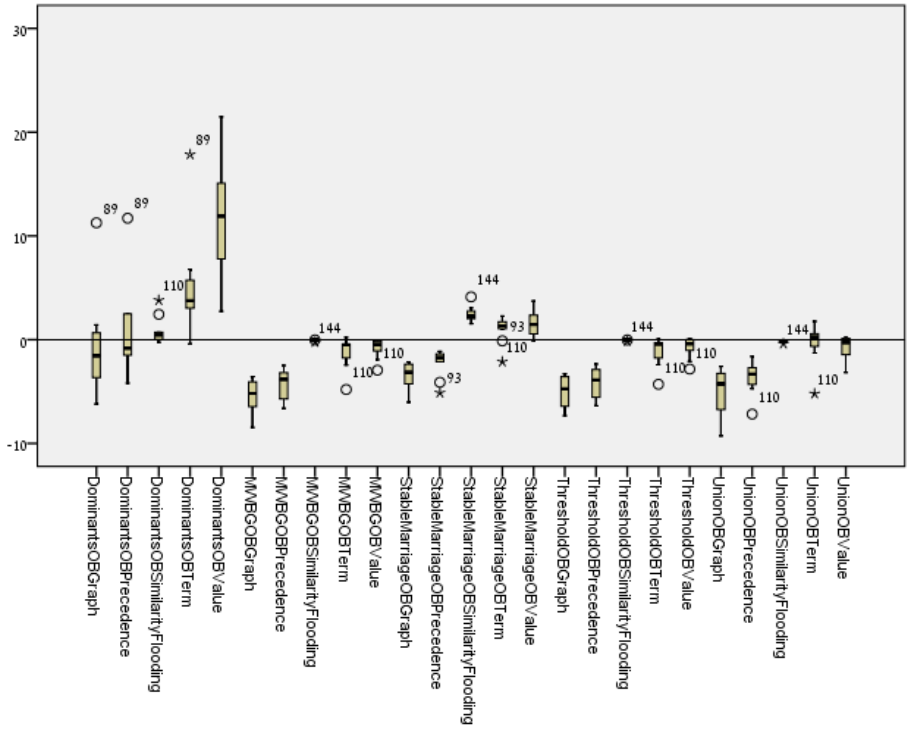


Fig. 5: Ontobuilder Pairs Box Plots

indicating that these two 1LMs provide noisy but rich results in which there is much information to be gained.

6 Related Work

The overwhelming majority of current schema matching evaluation measures are based upon *Precision* and *Recall*, first introduced to schema matching by Li and Clifton [14]. These measures and their derivatives have served the classic role of schema matching as a preceding step to mapping expression generation and served well in evaluating various matching systems. The emergence of new applications and research domains challenged their universal adequacy and spawned attempts to provide additional measures for additional needs. F-measure, borrowed from IR by Berlin and Motro [2], Error, borrowed from IR by Modica et. al. [19], and Information Loss, suggested by Mena et. al. [18] all provide methods to aggregate the results of Precision and Recall and evaluate algorithms on a single measure. Overall/Accuracy [17] and HSR [6] assume that schema matching is followed by a manual effort to validate correspondences and therefore are suggested as a better measure for *post-match effort*. A thorough comparison of the use of Precision, Recall, and their derivatives appears in [9], Ch. 3.4.

The above-mentioned measures all assume that evaluating schema matching entails comparing an absolute correspondence list with a similarly absolute *exact match*. Work done by Ehrig and Euzenat [8] progresses in the direction of relaxing this assumption by proposing additional alternative semantics to the set-inclusion semantics at the basis of classic Precision and Recall definitions. For example, to facilitate semantics of subsumption, a weight function is assigned to value tree distance between two terms of the source schema, where one was matched by the matcher and the other is the exact match. However, these semantics are still based upon valuation of binary relationships between the attributes matched. To the best of our knowledge, our work is the first to suggest a method for the assessment of non-binary matching results.

7 Conclusions

We have introduced non-binary schema matching evaluation, a tool in the hands of a matching system designer that can asses the quality of interim match results with no need for arbitrary match selection. We provide a formal model for schema matching evaluation using similarity spaces, offer three new non-binary measures and demonstrate how these new measures provide designers with capabilities that were previously unavailable.

In terms of future research we aim to continue and investigate different measures on various normed vector spaces. In addition, we intend to investigate the performance of different evaluation methods with respect to additional schema matching tasks. Finally, we shall also investigate non-binary reference vectors and the impact of using our new measures on schema matching problems where these vectors occur.

Acknowledgement The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under the NisB[2] project, grant agreement number 256955.

References

1. Algergawy, A., Nayak, R., and Saake, G. XML schema element similarity measures: a schema matching context. *On the Move to Meaningful Internet Systems: OTM 2009* (2009), 1246–1253.
2. Berlin, J., and Motro, A. Autoplex: Automated discovery of content for virtual databases. In *Cooperative Information Systems, 9th International Conference, CoopIS 2001* (Lecture Notes in Computer Science, September 5-7, 2001 2001), vol. 2172, Springer, pp. 108–122.
3. Bryant, V. *Metric spaces: iteration and application.* Cambridge Univ Pr, 1985.
4. Do, H. H., and Rahm, E. Coma: a system for flexible combination of schema matching approaches. In *Proceedings of VLDB* (2002), VLDB Endowment, pp. 610–621.

[2] http://nisb-project.eu/

5. Dong, X., Halevy, A., and Yu, C. Data integration with uncertainty. *The VLDB Journal 18* (2009), 469–500.
6. Duchateau, F. *Towards a generic approach for schema matcher selection: Leveraging user pre- and post-match effort for improving quality and time performance.* PhD thesis, Universite Monpelier II - Sciences et Techniques du Languedoc, 2009.
7. Duchateau, F., Bellahsene, Z., and Coletta, R. A flexible approach for planning schema matching algorithms. *On the Move to Meaningful Internet Systems: OTM 2008* (2008), 249–264.
8. Ehrig, M., and Euzenat, J. Relaxed precision and recall for ontology matching. In *Integrating Ontologies Workshop Proceedings* (2005), p. 25.
9. Gal, A. Uncertain schema matching. *Synthesis Lectures on Data Management 3*, 1 (2011), 1–97.
10. Gal, A., Martinez, M., Simari, G., and Subrahmanian, V. Aggregate query answering under uncertain schema mappings. In *ICDE* (2009), pp. 940–951.
11. Gal, A., Modica, G., Jamil, H., and Eyal, A. Automatic ontology matching using application semantics. *AI magazine 26*, 1 (2005), 21.
12. Gal, A., and Sagi, T. Tuning the ensemble selection process of schema matchers. *Information Systems 35*, 8 (2010), 845–859.
13. Lee, Y., Sayyadian, M., Doan, A. H., and Rosenthal, A. S. eTuner: tuning schema matching software using synthetic scenarios. *The VLDB journal 16*, 1 (2007), 97–122.
14. Li, W., and Clifton, C. Semint: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data and Knowledge Engineering 33*, 1 (2000), 49–84.
15. Mao, M., Peng, Y., and Spring, M. A harmony based adaptive ontology mapping approach. In *Proc. of SWWS* (2008).
16. Marie, A., and Gal, A. On the stable marriage of maximum weight royal couples. In *Proceedings of AAAI Workshop on Information Integration on the Web* (2007).
17. Melnik, S., Garcia-Molina, H., and Rahm, E. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *ICDE* (2002), IEEE, pp. 117–128.
18. Mena, E., Kashyap, V., Illarramendi, A., and Sheth, A. P. Imprecise answers in distributed environments: Estimation of information loss for multi-ontology based query processing. *Int. J. Cooperative Inf. Syst. 9*, 4 (2000), 403–425.
19. Modica, G., Gal, A., and Jamil, H. The use of machine-generated ontologies in dynamic information seeking. In *Cooperative Information Systems* (2001), pp. 433–447.
20. Tu, K., and Yu, Y. CMC: Combining multiple schema-matching strategies based on credibility prediction. In *Database Systems for Advanced Applications*, L. Zhou, B. Ooi, and X. Meng, Eds., vol. 3453 of *LNCS*. Springer Berlin / Heidelberg, 2005, pp. 995–995.
21. Zobel, J., and Moffat, A. Exploring the similarity space. *SIGIR Forum 32* (April 1998), 18–34.