Matching and Alignment: What is the Cost of User Post-match Effort ?*

(Short paper)

Fabien Duchateau[1] and Zohra Bellahsene[2] and Remi Coletta[2]

[1] Norwegian University of Science and Technology
NO-7491 Trondheim, Norway
fabiend@idi.ntnu.no
[2] LIRMM - Université Montpellier 2
161 rue Ada, 34392 Montpellier, France
{firstname.lastname}@lirmm.fr

Abstract. Generating new knowledge from scientific databases, fusioning products information of business companies or computing an overlap between various data collections are a few examples of applications that require data integration. A crucial step during this integration process is the discovery of correspondences between the data sources, and the evaluation of their quality. For this purpose, the *overall* metric has been designed to compute the post-match effort, but it suffers from major drawbacks. Thus, we present in this paper two related metrics to compute this effort. The former is called **post-match effort**, i.e., the amount of work that the user must provide to correct the correspondences that have been discovered by the tool. The latter enables the measurement of **human-spared resources**, i.e., the rate of automation that has been gained by using a matching tool.

1 Introduction

Data integration has now been studied for years, and many applications still make this research field an interesting challenge. Discovering correspondences between the data sources is one of the first steps of this integration process. As pointed out by [1], the quality obtained during this step mainly determines the quality of the whole data integration process. For this reason, matching communities (both schema and ontology) have been very prolific in producing matching tools during the last decades to automate the discovery of correspondences. Many surveys [2–5] and books [6, 7] reflect this interest.

To evaluate the results produced by their tools, these communities mainly use common quality metrics such as precision, recall, and F-measure. However, the aim of (semi-)automatic matching is to avoid a manual, labor and error-prone process. The post-match effort, which consists of checking the discovered

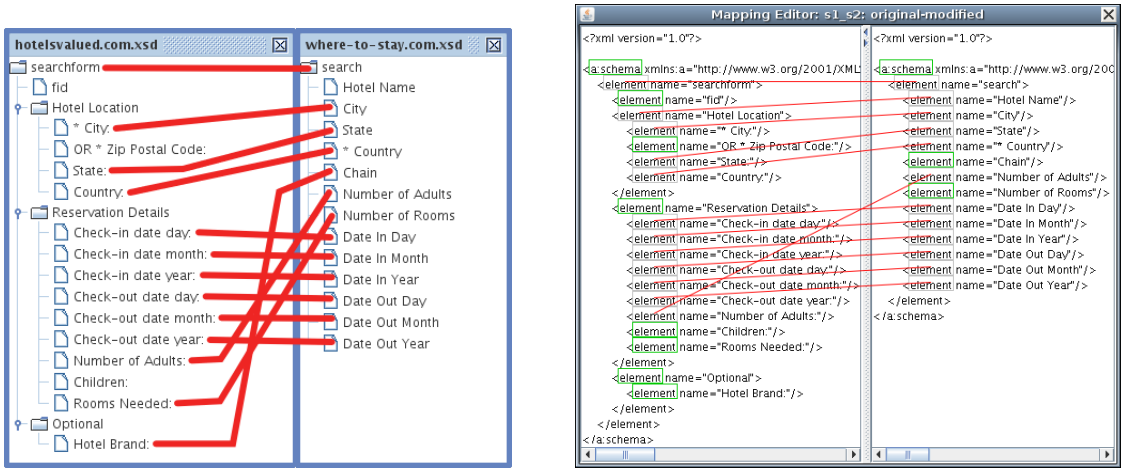
* Supported by ANR DataRing ANR-08-VERSO-007-04. The first author carried out this work during an ERCIM "Alain Bensoussan" Fellowship Programme.

correspondences and searching for the missing ones, should therefore be reduced at most. Yet, the available metrics hardly provide an estimation of this effort. F-measure is the harmonic mean between precision and recall, while it should add the correction cost of both measures. On the other hand, the overall (or accuracy) is a first attempt to evaluate the post-match effort [8].

Consequently, we propose a **post-match effort** measure and its inverse **human spared resources** which tackle these issues. It estimates the number of user interactions required to correct both precision and recall (i.e., to manually obtain a 100% F-measure). Thus, it takes into account the effort to (in)validate discovered correspondences, but also the search for missing ones between the data sources. This measure is sufficiently generic to be converted into the range $[0, 1]$ or in time units (e.g., seconds) and it does not require other specific inputs than those needed to compute precision, F-measure or overall.

2 Preliminaries

Correspondences are semantic links between elements of different data sources (schemas, ontologies) which represent the same real-world concept. Contrary to [9], evaluating the quality of the mapping (i.e., the transformation function between instances of one element into those of another element) is out of scope of this paper since we focus on correspondences. We also limit correspondences to *1:1* (i.e., one element is matched to only one element) or to *1:n* (i.e., one element is matched to several elements). Currently, only a few tools produce *n:m* correspondences. Figure 1(b) depicts an example of two schemas (from *hotel booking* web forms) and the correspondences discovered by a matching tool.



(a) Expert correspondences

(b) Correspondences discovered by a tool

Fig. 1. A running example : hotel booking webforms

A **matching dataset** is composed of a set of data sources (schemas, ontologies) to be matched and the set of expert correspondences. This set of expert correspondences is considered as complete and trustful. Such datasets, also called testbeds or test collections are used by most evaluation tools as an oracle, against which they can compare different approaches or tools. To evaluate the matching quality, three measures are commonly accepted in the literature. **Precision** calculates the proportion of correct correspondences extracted among the discovered ones. Another typical measure is **recall** which computes the rate of correct discovered correspondences among all correct ones. **F-measure** is a trade-off between precision and recall. We propose to complete these measures with our post-match effort measure, which is presented in the next section.

3 Post-match Effort Metric

We present in this paper two related metrics: **post-match effort** and **human-spared resources**.

3.1 Intuition and Running Example

A set of discovered correspondences, provided by a schema matching tool, has two issues, namely (i) incorrect discovered correspondences and (ii) missing (correct) correspondences. Users first have to check each correspondence from the set, either to (in)validate or complete it (in case of 1:n correspondences). Then, they have to browse the schemas and discover the missing correspondences. Thus, we propose to evaluate this user post-match effort by **computing the number of user interactions** to reach a 100% F-measure, i.e., to correct the two previously mentioned issues. A user interaction is an (in)validation of one pair of schema elements (either from the set of discovered correspondences or between the schemas). We first introduce three assumptions which underlie our metric:

- **Worst case**, which means that all pairs of schema elements, which have not already been matched, must be (in)validated. In addition, the last (in)validated pair would be a correspondence.
- **Uniformity**, i.e., missed correspondences are discovered with the same frequency (and not at random). Although not realistic, this assumption mainly enables a fair comparison when evaluating the post-match effort for different tools. The worst case assumption anyhow guarantees that the last validated pair is a correct correspondence.
- Only **correspondences 1:1** are taken into account. The metric can be applied with *1:n correspondences* (represented by several 1:1 correspondences), but we do not consider more complex correspondences (namely *n:m*). However, we note that a post-processing technique could transform the 1:1 validated correspondences into complex correspondences by relying on the data.

Now, let us introduce an example. Figure 1(b), presented in the preliminaries section, depicts a set of correspondences discovered by a matching tool between

two hotel booking schemas. The expert set of correspondences is shown by figure 1(a). We notice that one discovered correspondence is incorrect: *(Hotel Location, Hotel Name)*. Consequently, it has to be invalidated. Besides, the matching tool has missed two correspondences, namely *(Hotel Brand:, Chain)* and *(Rooms Needed:, Number of Rooms)*. These two correspondences have to be searched among the 23 pairs that have not been validated (8×3 possible pairs minus 1 incorrect pair discovered by the tool).

3.2 Estimating the Number of User Interactions

We define the number of user interactions as a positive number which represents the number of user interactions to obtain a 100% F-measure from a set of discovered correspondences. It consists of two steps which are described below.

Given two schemas S_ℓ and S_L of respective sizes $|S_\ell|$ and $|S_L|$, with $|S_\ell| \leq |S_L|$ (i.e., S_L is a larger schema than S_ℓ), their expert set of correspondences E contains $|E|$ correspondences. A matching tool applied against these schemas has discovered a set of correspondences M, which contains $|M|$ correspondences. Among these discovered correspondences, $|R|$ of them are correct, with $0 \leq |R| \leq |M|$. To compute the number of user interactions, only the five inputs $|S_\ell|, |S_L|, |E|, |M| and |R|$ are required. In our example, we have the following values:

- $|S_\ell| = 14$, the number of elements in the smallest schema[3].
- $|S_L| = 19$, the number of elements in the largest schema[3].
- $|E| = 13$, the number of expert correspondences
- $|M| = 12$, the number of correspondences discovered by the matching tool, shown in figure 1(b).
- $|R| = 11$, the number of correct correspondences discovered by the matching tool.

Step 1: checking of all discovered correspondences. This step is very easy to compute. A user has to check each correspondence from the set of discovered correspondences, and (in)validate it. Thus, this requires a number of interactions equal to the number of discovered correspondences in the set, $|M|$ in our case. We call this metric effort_{prec} since it is directly impacted by precision. Indeed, a high precision reduces the number of user interactions since there are fewer incorrect correspondences which have been discovered. Note that at the end of this step, the precision value is equal to 100%.

$$\text{effort}_{prec} = |M| \tag{1}$$

In our example, there are 12 discovered correspondences, thus $\text{effort}_{prec} = 12$. It means that the number of user interactions during this step is equal to 12, among which 11 validations and 1 invalidation for the incorrect correspondence.

Step 2: manual discovery of missed correspondences. The second step deals with the manual discovery of all missing correspondences. At the end of this

[3] We do not count the root element tagged with <*a:schema*>.

step, recall reaches 100%, and F-measure too. We assume that all pairs which have not been invalidated yet must be analyzed by the user. As we consider only 1:1 correspondences, elements that have already been matched are not checked anymore. The main idea is to check every unmatched element from the smallest schema against all unmatched elements from the largest schema.

Due to the uniformity assumption, we manually discover a missing correspondence with the same frequency. This frequency is computed by dividing the number of unmatched elements in the smallest schema by the number of missing correspondences, as shown by Formula 2. Thanks to 1:1 correspondences assumption, the number of correct correspondences $|R|$ is at most equal to the number of correctly matched elements in each schema (i.e., $0 \leq |R| \leq |S_\ell|$). Hence we can compute $|S_\ell| - |R|$.

$$\text{freq} = \frac{|S_\ell| - |R|}{|E| - |R|} \tag{2}$$

Back to our example, $\text{freq} = \frac{14-11}{13-11} = \frac{3}{2}$ means that the user will manually find a missing correspondence for every three unmatched elements from the smallest schema.

Since we now know the frequency, we can compute the number of interactions using a sum function. We call this metric effort_{rec} since it is affected by recall. The higher recall you achieved, the fewer interactions you require during this step. $|S_L| - |R|$ denotes the number of unmatched elements from the largest schema. With i standing for the analysis of the i^{th} unmatched element from S_ℓ, $\frac{i}{\text{freq}}$ represents the discovery of a missing correspondence (when it reaches 1). We also uniformly remove the pairs which may have been already invalidated during step 1, by computing $\frac{|M|-|R|}{|S_\ell|-|R|}$. Thus, we obtain this Formula 3:

$$\text{effort}_{rec} = \sum_{i=1}^{|S_\ell|-|R|} (|S_L| - |R| - \frac{i}{\text{freq}} - \frac{|M| - |R|}{|S_\ell| - |R|}) \tag{3}$$

To sum up, for each unmatched of the smallest schema, the user has to analyze all elements of the largest schema, except for those already matched ($|R|$), those already part of a match previously discovered ($\frac{i}{\text{freq}}$) and those invalidated during the first step ($\frac{|M|-|R|}{|S_\ell|-|R|}$). We now detail for our example the successive iterations of this sum function, which vary from 1 to 3.

- $\textit{effort}_{rec}(i = 1)$, $19 - 11 - \frac{1}{1.5} - \frac{1}{3} = 7$
- $\textit{effort}_{rec}(i = 2)$, $19 - 11 - \frac{2}{1.5} - \frac{1}{3} = 6\frac{1}{3}$
- $\textit{effort}_{rec}(i = 3)$, $19 - 11 - \frac{3}{1.5} - \frac{1}{3} = 5\frac{2}{3}$

Thus, the second step to discover all missing correspondences requires $\textit{effort}_{rec}$ $= 7 + 6\frac{1}{3} + 5\frac{2}{3} = 19$ user interactions.

Finally, to compute the number of user interactions between two schemas S_ℓ and S_L, noted *nui*, we need to sum the values of the two steps, thus resulting in Formula 4. If the set of correspondences is empty, then using a matching tool

was useless and the number of user interactions is equal to the number of pairs between the schemas.

$$\text{nui}(S_\ell, S_L) = \begin{cases} |S_\ell| \times |S_L| & if |M| = 0 \\ \\ \text{effort}_{prec} + \text{effort}_{rec} & otherwise \end{cases} \quad (4)$$

In our example, the user needs a number of user interactions $nui = 12+19 = 31$ to correct the set of correspondences produced by the tool.

3.3 Normalization and Generalization

The number of user interactions is not sufficient to measure the benefit of using a matching tool. Indeed, a given number of interactions may appear as an incredible effort for correcting the set of correspondences of two small data sources, but it may seem acceptable when dealing with large data sources. Thus, our post-match effort (and its inverse, human spared resources) is a normalization of this number of user interactions based on the size of the data sources. Then, we explain how to generalize the post-match effort when there are more than two data sources.

Normalization. From the number of user interactions, we can normalize the **post-match effort** value into [0,1]. It is given by Formula 5. Indeed, we know the number of possible pairs ($|S_\ell| \times |S_L|$). Checking all these pairs means that the user performs a manual matching, nui = $|S_\ell| \times |S_L|$ and pme = 100%.

$$\text{pme}(S_\ell, S_L) = \frac{\text{nui}(S_\ell, S_L)}{|S_\ell| \times |S_L|} \quad (5)$$

We can also compute the percentage of automation of the matching process thanks to a matching tool. This metric, noted *hsr*, for **human spared resources**, is given by Formula 6. This measure enables the computation of the rate of automation by the matching process.

$$\text{hsr}(S_\ell, S_L) = 1 - \frac{\text{nui}(S_\ell, S_L)}{|S_\ell| \times |S_L|} = 1 - \text{pme}(S_\ell, S_L) \quad (6)$$

If a matching tool achieves a 20% post-match effort, this means that the user has to perform a 20% manual matching for removing and adding correspondences, w.r.t. a complete (100%) manual matching. Consequently, we can deduce that the matching tool managed to automate 80% of the matching process. In our dating example, the post-match effort is equal to pme = $\frac{31}{14\times19} \simeq 12\%$ and human spared resources is equal to hsr = $1 - 0.12 \simeq 88\%$. The matching tool has spared 88% resources of the user, who still has to manually perform 12% of the matching process.

Generalization. As matching scenarios may contain more than two data sources, we need to generalize the post-match effort formula. Let us consider that a matching scenario contains n data sources such as a set $< S_1, S_2, ..., S_n >$. The generalized post-match effort, noted pme_{gen}, is given by Formula 7. It is the

sum of all numbers of user interactions in all possible couples of data sources, divided by the sum of all numbers of pairs in all possible couples of data sources.

$$\text{pme}_{gen} = \frac{\sum_{i=1}^{i=n} \sum_{j=i+1}^{j=n} \text{nui}(S_i, S_j)}{\sum_{i=1}^{i=n} \sum_{j=i+1}^{j=n} |S_i| \times |S_j|} \tag{7}$$

4 Related Work

To the best of our knowledge, the overall measure (also named accuracy in [8]) is the only one to compute a post-match effort [10]. It is computed with the following formula in the range $[-\infty, 1]$:

$$Overall = Recall \times \left(2 - \frac{1}{Precision}\right) \tag{8}$$

A major drawback of this measure deals with the fact that removing irrelevant correspondences is considered as difficult (in terms of user effort) as adding missed correspondences. However, this is rarely the case in real-world scenarios. Another drawback explained by the authors deals with a precision below 50%: it implies more effort from the user to remove extra correspondences and add missing ones than to manually do the matching, thus resulting in a negative overall value which is often disregarded. On the contrary, our measure returns values in the range $[0, 1]$ and it does not assume that a low precision involves much effort during post-match. Finally, the overall measure does not consider the size of the data sources. Yet, even with the same number of expert correspondences, the manual task for checking the discovered correspondences and finding the missed correspondences in two large data sources requires a larger effort than in small data sources. To sum up this comparison, overall is mainly more pessimistic than HSR. This is illustrated by Figure 2 which depicts the values of overall and HSR when the number of discovered correspondences ($|M|$) and the number of correct discovered correspondences ($|R|$) vary. In this plot, two parameters are fixed: the number of expert correspondences $|E|$ to 200 and the average size of the data sources $|S|$ to 500. All overall values are less than 0 when the number of correct discovered correspondences is at most half of the number of discovered correspondences, which is an obvious limitation. We also notice that overall may be more optimistic than HSR with high precision and recall values (in our plot, when $|M|$ and $|R|$ are close to $|E|$). The reason deals with the size of the data sources, which is not considered by overall. These comments are verified with other values of $|E|$ and $|S|$. Due to page limit, all plots are available online[4].

5 Conclusion

In this paper, we have presented a former metric which computes the post-match effort while the latter estimates the percentage of automation due to the use of a

[4] Appendix at http://november.idi.ntnu.no/∼fabien/appendixCoopis11.pdf

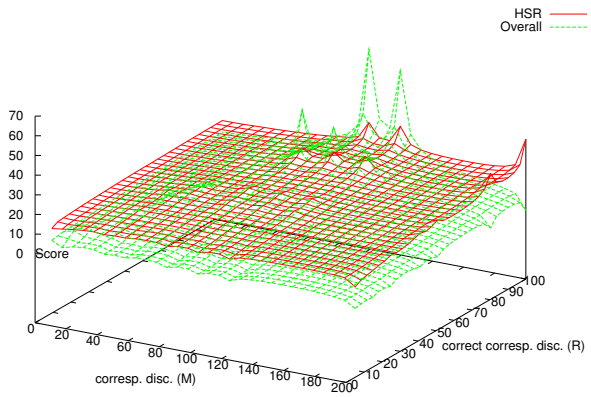


Fig. 2. A Comparison of Overall and HSR

matching tool. The scores computed by our measures and presented as a number of user interactions can be converted in time units. In addition, except for the size of the data sources, computing these metrics does not require more information than traditional quality measures. As a future work, we first intend to extend our measure so that it takes into account the top-K correspondences returned by several matching tools. Then, we would like to quantify pre-match effort too.

References

1. Smith, K., Morse, M., Mork, P., Li, M., Rosenthal, A., Allen, D., Seligman, L.: The role of schema matching in large enterprises. In: CIDR. (2009)
2. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal **10**(4) (2001) 334–350
3. Yatskevich, M.: Preliminary evaluation of schema matching systems. Technical Report DIT-03-028, Informatica e Telecomunicazioni, University of Trento (2003)
4. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Journal of Data Semantics IV (2005) 146–171
5. Noy, N.F., Doan, A., Halevy, A.Y.: Semantic integration. AI Magazine **26**(1) (2005) 7–10
6. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag (2007)
7. Bellahsene, Z., Bonifati, A., Rahm, E.: Schema Matching and Mapping. Springer-Verlag, Heidelberg (DE) (2011)
8. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: ICDE. (2002) 117–128
9. Alexe, B., Tan, W.C., Velegrakis, Y.: STBenchmark: towards a benchmark for mapping systems. Proceedings of the VLDB **1**(1) (2008) 230–244
10. Do, H.H., Melnik, S., Rahm, E.: Comparison of schema matching evaluations. In: Web, Web-Services, and Database Systems Workshop. (2002)