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

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, a few measures such as *precision* and *recall* have been designed. However, these measures do not evaluate user post-match effort, that matching approaches aim at reducing. Furthermore, the *overall* metric suffers from a major drawback. Thus, we present in this paper two measures to compute this user effort during the post-match phase, that takes into account both the correction of discovered correspondences and the manual search for missing ones. A set of experiments with three matching tools including a comparison with the *overall* measure highlights the benefits of our metrics. We also show that time performance during matching is not significant w.r.t. time performance during post-matching.

1 Plots : Evaluation

In this section, we compare the quality results of three matching tools to demonstrate the benefits of our measure.

1.1 Evaluation Protocol

The evaluation is performed with three matching tools and six datasets. More specifically, the following matching tools are able to match schemas or ontologies: COMA++ [1, 2], Similarity Flooding (SF) [3, 4] and YAM [5, 6]. **COMA++** is a hybrid matching tool whose main goal is to build a matrix of similarity values for all pairs of elements and similarity measures. Different strategies can be

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

applied to this matrix for detecting the correspondences to be displayed to the user. **Similarity Flooding / Rondo** is also an hybrid matcher that discovers correspondences in two phases: an initial element-level terminological matching refined by a structural similarity measure (propagation of similarity values of the neighbors). **YAM (Yet Another Matcher)** is not (yet) another matching system as it enables the generation of *à la carte* matchers for a given matching scenario. Based on the user requirements, YAM learns how to best apply both similarity measures and classifiers in concert to achieve the best matching quality. Further details about other matching approaches, both in ontology alignment and schema matching, are given in these books and surveys [7–13].

We report results by dataset. Table 1 sums up the properties of datasets. *Label heterogeneity* is computed by terminological similarity measures applied to the expert set of correspondences. If these measures are able to discover most of the correspondences, this means that the labels have a very low heterogeneity. Conversely, if terminological measures only discover a few correspondences, then the labels are strongly heterogeneous. *Domain specific* means that the vocabulary is uncommon and it cannot be found in general dictionaries like Wordnet[3].

	Label heterogeneity		**Domain**	**Average Size**			**Structure**	
	Low (or normalized)	Average	**Specific**	Small (<10)	Average (10-100)	Large (>100)	Flat	Nested (3<depth<7)
Betting		x			x		x	
Currency		x			x			x
Finance		x	x		x		x	
Order	x					x		x
Travel		x		x			x	
Univ. courses		x			x		x	

Table 1. Properties of the datasets

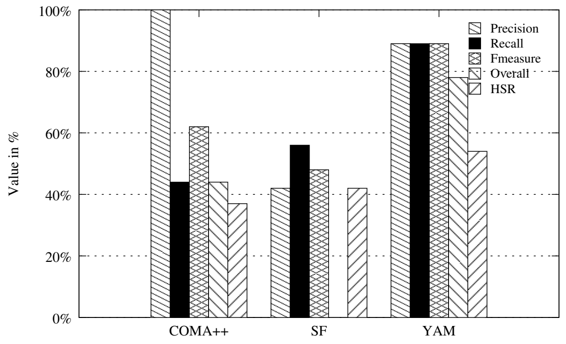
1.2 Experimental Results

In the following quality plots, five measures are shown: precision, recall, F-measure, overall and human spared resources (HSR). The overall values have been limited to 0 instead of -∞. One should consider a negative overall value as not significant as it was explained in [3]. *HSR* is the reverse of post-match effort, i.e., the percentage of automation due to the use of the matching tool.

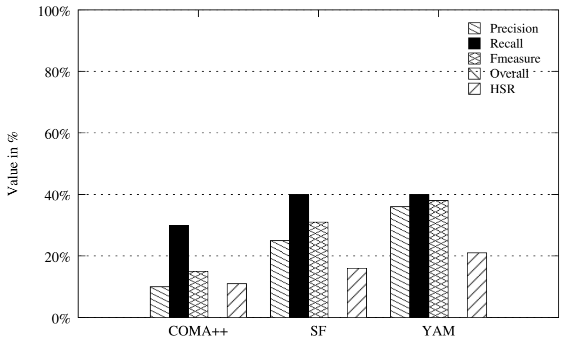
Betting dataset. The betting dataset is provided by authors of [14]. It contains two flat ontologies gathered from betting websites. Figure 1(a) depicts the quality of the matching tools for this dataset. A first comment deals with the fact that a high precision tends to promote a high overall (e.g., COMA++). Although YAM achieves a 80% F-measure, it achieves a low HSR (54%) because the number of elements involved in a correspondence is not important w.r.t. the total number of elements in the ontologies. Consequently, checking if no

[3] http://wordnet.princeton.edu

correspondence has been forgotten (step 2 of our HSR measure) requires many user (in)validations.



(a) Quality for the *betting* dataset



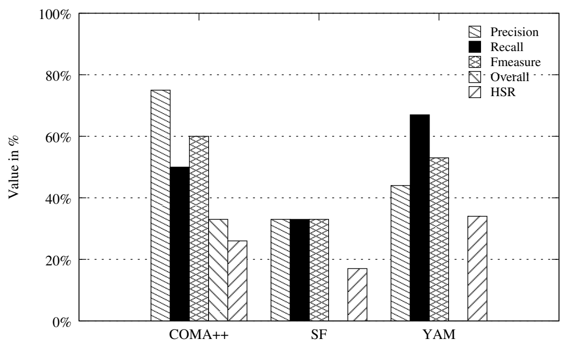
(b) Quality for the *order* dataset

Order dataset. This dataset deals with large business schemas. The first one is drawn from the XCBL collection[4] while the second schema is extracted from OAGI collection[5]. The labels of the schema elements are normalized. As depicted by figure 1(b), we notice that all matching tools obtain low results (F-measures less than 30%). Contrary to what overall values (all equal to 0) may suggest, all tools have discovered a few correct correspondences. Thus, they enable to spare some resources (HSR between 11% and 20%). Although SF and YAM obtain the same recall, YAM achieves a higher precision, which directly affects HSR. This is because the set of discovered correspondences is large, and the number of (in)validations for this step is therefore reduced with a better precision.

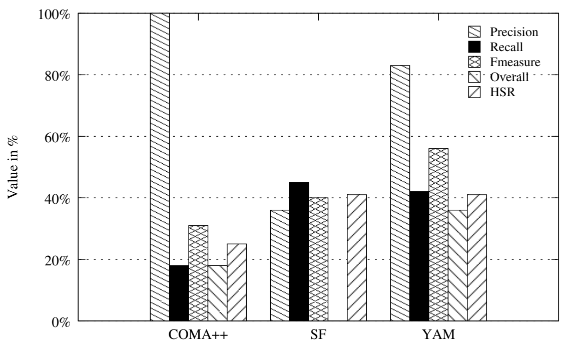
Currency dataset. The currency dataset gathers two popular web services[6]. Figure 1(c) depicts the quality obtained by the matching tools. It shows how overall is strongly correlated with precision. Let us analyze the results of COMA++ and YAM. The first one has a high precision and a lower recall (75% and 50%), thus resulting in a 34% overall. YAM achieves a higher recall than precision (44% and 67%), resulting a in negative overall value ! On the contrary, our HSR measure returns results around 30% for both tools, meaning that their use enabled some resources sparing.

Finance dataset. The finance dataset has been extracted from various websites by the authors of [14]. Its two ontologies own a specific vocabulary. On figure 1(d), we notice that YAM and SF have a roughly similar recall and the same HSR values. Yet, YAM achieves a higher precision than SF (resp. 82% and 38%). We explain this by the fact that the sets of discovered correspondences (both

[4] http://www.xcbl.org
[5] http://www.oagi.org
[6] http://www.seekda.com



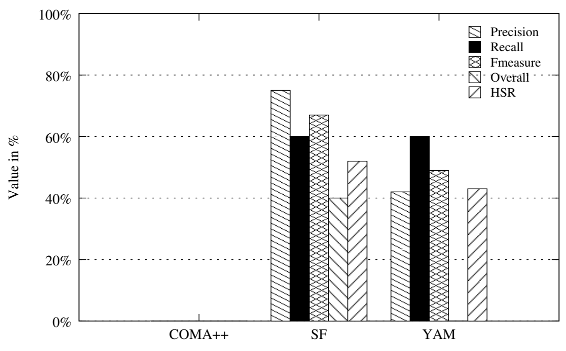
(c) Quality for the *currency* dataset



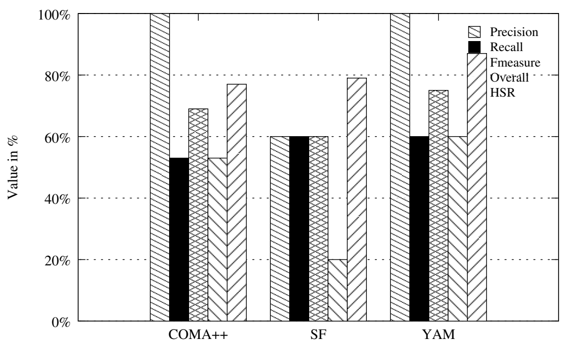
(d) Quality for the *finance* dataset

for YAM and SF) have a small size (about ten elements). Consequently, the (in)validation of these discovered correspondences only requires a few interactions with a limited impact on HSR.

Travel dataset. The travel dataset includes schemas that have been extracted from airfare web forms[7]. The quality of the matching process is depicted by Figure 1(e). COMA++ did not discovered any correspondences. Consequently, its overall and HSR are equal to 0. Maybe the default threshold applied to computed similarity values is too high for this dataset. Although YAM and SF achieve the same recall, SF obtains a positive overall (40%) since its precision is 74%. On the contrary, YAM, with a precision equal to 40%, has a negative overall. Our HSR measure computes more realistic results (between 40% to 50%).



(e) Quality for the *travel* dataset



(f) Quality for the *univ-courses* dataset

Univ-courses dataset. This last experiment utilizes two schemas from the Thalia collection [15]. As shown on figure 1(f), the interesting point about

[7] The UIUC web integration repository, http://metaquerier.cs.uiuc.edu/repository

these results deals with the HSR, which may be superior to F-measure (e.g., COMA++). Let us first compare YAM and SF. Both have the same recall (60%) but YAM's precision is improved by 40%. Thus, this means that the difference of 8% in their respective HSR is only for correcting precision. On the contrary, YAM and COMA++ achieves the same precision (100%) but YAM discovered more correct correspondences (recall improved by 7%). The difference of 10% in their respective HSR is only for correcting recall. As COMA++ and SF have roughly the same HSR, we can therefore deduce that for this dataset, the cost for removing the 40% of incorrect correspondences (SF) is similar to the cost for finding the 7% of missed correspondences (COMA++). This clearly highlights that in most datasets, the cost for discovering missed correspondences is higher than the one to invalidate incorrect discovered correspondences.

1.3 Discussion about the Evaluation

We finally discuss the results of these experiments.

– Our HSR measure is more realistic than overall. When precision is below 50%, the overall values are negative. Yet, it does not mean that the value is insignificant, as stated in [3], and that using the tool was a lack of time. HSR is somehow more correlated to recall than to precision. With a high precision and an average recall, HSR does not reach high values. This is due to the fact that a low recall implies more costly post-match effort from the user than precision does. Thus, HSR is a more balanced metric than overall, and probably more realistic as well.
– The evaluated tools mainly promote precision to the detriment of recall. Besides, this choice strongly impacts post-match effort. It could be smarter to promote recall, for instance with datasets involving large data sources.
– Table 2 presents the time performance of the matching tools for all datasets. Here, we underline the fact that time performance during matching is nowadays not much significant[8]. Indeed, matching time does not exceed one minute to match any of these datasets. Conversely, the time during post-match effort is crucial. Based on the Number of User Interactions *NUI* (required to compute HSR or PME), we have converted the post-match effort into time. Indeed, we assume that the user needs 30 seconds to (in)validate a pair of elements. Thanks to state-of-the-art GUI like [16], this assumption is quite realistic. Except for some small datasets (travel and univ-courses), the post-match effort requires several hours from the user. Thus, measuring this effort is necessary to improve it.

[8] Except in some specific environments, for instance large scale and highly dynamic.

	COMA++			Similarity Flooding			YAM		
	match	NUI	post-match	match	NUI	post-match	match	NUI	post-match
Betting	1 sec	362	3 hours	1 s	333	2.5 hours	1 s	264	2 hours
Currency	5 sec	247	2 hours	1 s	272	2 hours	1 s	235	2 hours
Finance	1 sec	439	3.5 hours	1 s	360	3 hours	1 s	352	3 hours
Order	43 sec	11366	94.5 hours	2 s	10934	91 hours	12 s	10941	91 hours
Travel	1 sec	78	39 min	1 s	47	23 min	1 s	47	23 min
Univ. courses	1 sec	67	33 min	1 s	56	28 min	1 s	37	18 min

Table 2. Time performance on the different datasets

2 Plots : Comparison of Overall and HSR

The second part of this appendix includes five plots which depicts a comparison of overall and HSR for different values of S (average size of the data sources), E (number of expert correspondences), M (number of correspondences discovered by a tool), and R (number of correct correspondences discovered by a tool).

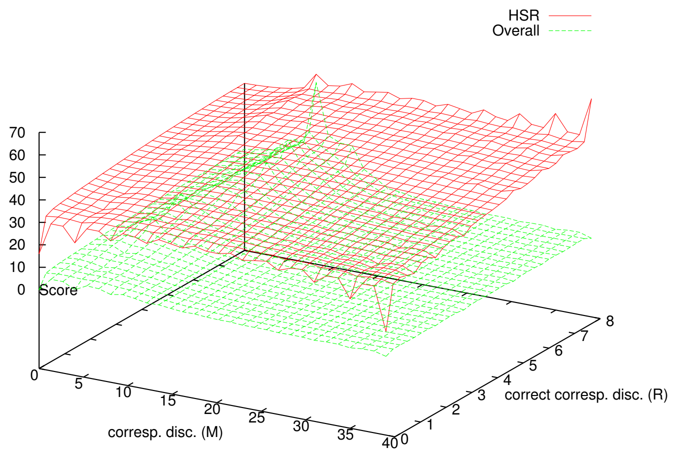


Fig. 1. Comparison of overall and HSR for two data sources of 20 elements and 8 expert correspondences

References

1. Do, H.H., Rahm, E.: COMA - A System for Flexible Combination of Schema Matching Approaches. In: VLDB. (2002) 610–621
2. Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++. In: ACM SIGMOD. (2005) 906–908

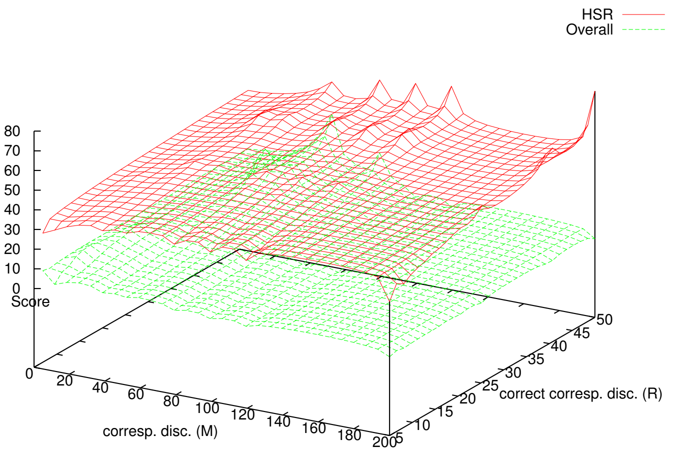


Fig. 2. Comparison of overall and HSR for two data sources of 100 elements and 50 expert correspondences

3. 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
4. Melnik, S., Rahm, E., Bernstein, P.A.: Developing metadata-intensive applications with rondo. J. of Web Semantics **I** (2003) 47–74
5. Duchateau, F., Coletta, R., Bellahsene, Z., Miller, R.J.: (Not) Yet Another Matcher. In: CIKM. (2009) 1537–1540
6. Duchateau, F., Coletta, R., Bellahsene, Z., Miller, R.J.: YAM: a schema matcher factory. In: CIKM. (2009) 2079–2080
7. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (DE) (2007)
8. Bellahsene, Z., Bonifati, A., Rahm, E.: Schema Matching and Mapping. Springer-Verlag, Heidelberg (DE) (2011)
9. Euzenat, J., et al.: State of the art on ontology matching. Technical Report KWEB/2004/D2.2.3/v1.2, Knowledge Web (2004)
10. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal **10**(4) (2001) 334–350
11. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Journal of Data Semantics IV (2005) 146–171
12. Yatskevich, M.: Preliminary evaluation of schema matching systems. Technical Report DIT-03-028, Informatica e Telecomunicazioni, University of Trento (2003)
13. Noy, N.F.: Semantic integration: A survey of ontology-based approaches. ACM SIGMOD Record **33**(4) (2004) 65–70
14. Marie, A., Gal, A.: Boosting schema matchers. In: OTM '08, Berlin, Heidelberg, Springer-Verlag (2008) 283–300

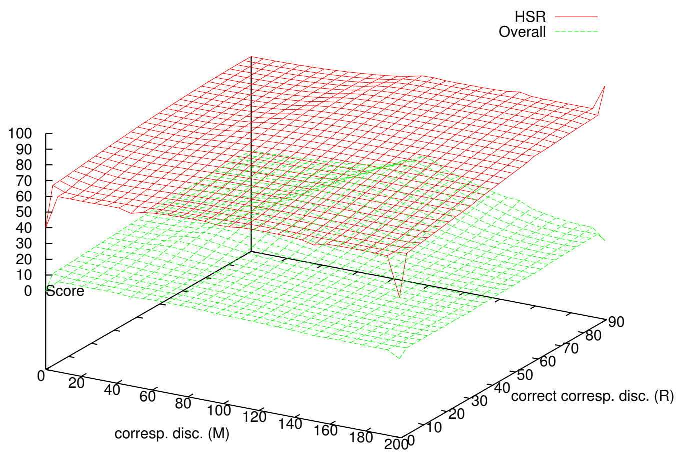


Fig. 3. Comparison of overall and HSR for two data sources of 100 elements and 90 expert correspondences

15. Hammer, J., Stonebraker, M., , Topsakal, O.: Thalia: Test harness for the assessment of legacy information integration approaches. In: Proceedings of ICDE. (2005) 485–486
16. Cruz, I.F., Sunna, W., Makar, N., Bathala, S.: A visual tool for ontology alignment to enable geospatial interoperability. J. Vis. Lang. Comput. **18**(3) (2007) 230–254

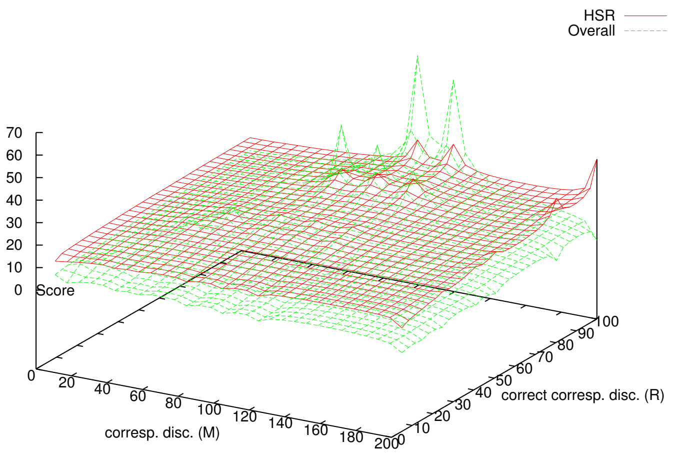


Fig. 4. Comparison of overall and HSR for two data sources of 500 elements and 100 expert correspondences

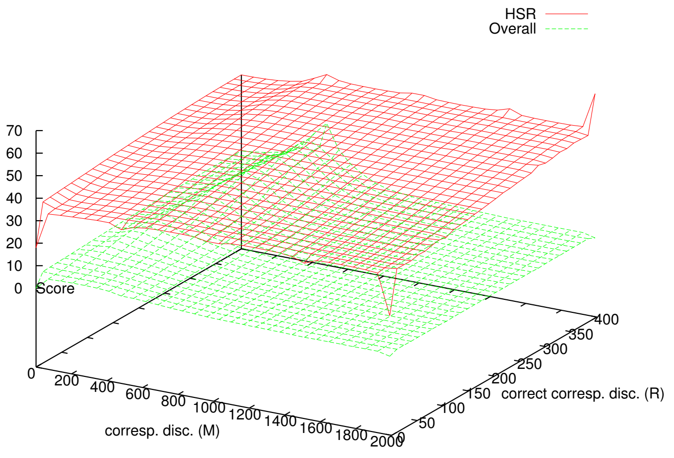


Fig. 5. Comparison of overall and HSR for two data sources of 1000 elements and 400 expert correspondences