Measuring the Quality of an Integrated Schema*

Fabien Duchateau[1] and Zohra Bellahsene[2]

[1] CWI, Science Park 123
1098 XG Amsterdam, The Netherlands
fabien@cwi.nl
[2] LIRMM - Université Montpellier 2
161 rue Ada, 34392 Montpellier, France
bella@lirmm.fr

Abstract. Schema integration is a central task for data integration. Over the years, many tools have been developed to discover correspondences between schemas elements. Some of them produce an integrated schema. However, the schema matching community lacks some metrics which evaluate the quality of an integrated schema. Two measures have been proposed, completeness and minimality. In this paper, we extend these metrics for an expert integrated schema. Then, we complete them by another metric that evaluates the structurality of an integrated schema. These three metrics are finally aggregated to evaluate the proximity between two schemas. These metrics have been implemented as part of a benchmark for evaluating schema matching tools. We finally report experiments results using these metrics over 8 datasets with the most popular schema matching tools which build integrated schemas, namely COMA++ and Similarity Flooding.

1 Introduction

Schema integration is the process of merging existing data sources schemas into one unified schema named global schema or integrated schema. This unified schema serves as a uniform interface for querying the data sources [1]. However, integrated schema can also serve in many other applications. Indeed, due to growing availability of information in companies, agencies, or on the Internet, decision makers may need to quickly understand some concepts before acting, for instance for building communities of interest [2]. In these contexts, the quality of an integrated schema is crucial both for improving query execution through the mediated schema and for data exchange and concepts sharing [3]. Although schema matching tools mainly emphasize the discovering of correspondences, most of them also generate an integrated schema based on these correspondences. Evaluating the quality of discovered correspondences is performed by

* 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.

J. Parsons et al. (Eds.): ER 2010, LNCS 6412, pp. 261–273, 2010.

using widely accepted measures (precision and recall). Yet, the schema matching community lacks some measures for assessing the quality of the integrated schema that is also automatically produced by the tools.

Consequently, authors of [4] have proposed the completeness and minimality measures. The former represents the percentage of data sources concepts which are covered by the integrated schema, while the latter checks that no redundant concept appears in the integrated schema. As stated by Kesh [5], these metrics are crucial to produce a more efficient schema, i.e. that reduces query execution time. However, they do not measure the quality of the structure of the produced integrated schema. We believe that the structure of an integrated schema produced by a schema matching tool may also decrease schema efficiency if it is badly built. Besides, an integrated schema that mainly preserves the semantics of the source schemas is easier to interpret and understand for an end-user.

This paper discusses the evaluation of the quality for integrated schemas. First, we adapt **completeness** and **minimality**, proposed in [4], for an expert integrated schema. Then, we complete them by another metric that evaluates the **structurality** of an integrated schema. These three metrics are finally aggregated to evaluate the **schema proximity** of two schemas. Experiments using two state-of-the-art schema matching tools enable us to demonstrate the benefits of our measures.

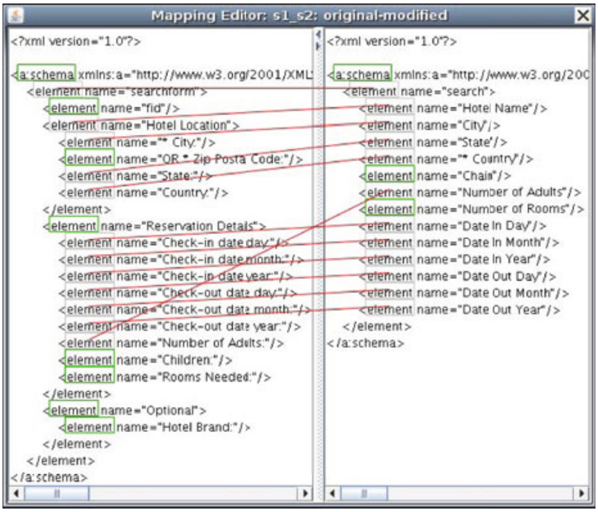
The rest of the paper is organised as follows: first, we give some definitions in Section 2. Section 3 covers the new measures we have designed for evaluating integrated schemas. We report in Section 4 the results of two schema matching tools. Related work is presented in Section 5. Finally, we conclude and outline future work in Section 6.

2 Preliminaries

Here we introduce the notions used in the paper. **Schema matching** is the task which consists of discovering semantic correspondences between schema elements. We consider **schemas** as edge-labeled trees (a simple abstraction that can be used for XML schemas, web interfaces, or other semi-structured or structured data models). **Correspondences** (or **mappings**) are links between schema elements which represent the same real-world concept. We limit correspondences to *1:1* (i.e., one schema element is matched to only one schema element) or to *1:n* (i.e., one schema element is matched to several schema elements). Currently, only a few schema matching tools produce *n:m* correspondences. Figure 1 depicts an example of two schemas (from *hotel booking* web forms) and the correspondences discovered by a schema matching tool.

A **schema matching dataset.** is composed of a schema matching scenario (the set of schemas to be matched), the set of expert mappings (between the schemas of the scenario) and the integrated expert schema.

A metric proposed in this paper uses a **rooted Directed Acyclic Graph** (rDAG) for evaluating the schema structure. Schemas can be seen as rDAGs. A rDAG is a DAG, expressed by a triple $< V, E, r >$ where:



```
<?xml version="1.0"?>

<a:schema xmlns:a="http://www.w3.org/2001/XML
  <element name="searchform">
    <element name="fid"/>
    <element name="Hotel Location">
      <element name="* City:"/>
      <element name="OR * Zip Posta Code:"/>
      <element name="State:"/>
      <element name="Country:"/>
    </element>
    <element name="Reservation Details">
      <element name="Check-in date day:"/>
      <element name="Check-in date month:"/>
      <element name="Check-in date year:"/>
      <element name="Check-out date day:"/>
      <element name="Check-out date month:"/>
      <element name="Check-out date year:"/>
      <element name="Number of Adults:"/>
      <element name="Children:"/>
      <element name="Rooms Needed:"/>
    </element>
    <element name="Optional">
      <element name="Hotel Brand:"/>
    </element>
  </element>
</a:schema>
```

```
<?xml version="1.0"?>

<a:schema xmlns:a="http://www.w3.org/200
  <element name="search">
    <element name="Hotel Name"/>
    <element name="City"/>
    <element name="State"/>
    <element name="* Country"/>
    <element name="Chain"/>
    <element name="Number of Adults"/>
    <element name="Number of Rooms"/>
    <element name="Date In Day"/>
    <element name="Date In Month"/>
    <element name="Date In Year"/>
    <element name="Date Out Day"/>
    <element name="Date Out Month"/>
    <element name="Date Out Year"/>
  </element>
</a:schema>
```

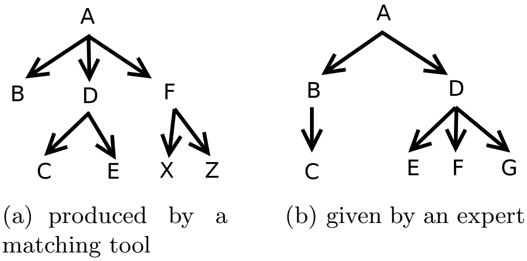
Fig. 1. Correspondences between two *hotel booking* schemas

- V is a set of elements, noted $V =< e_0, e_1, ..., e_n >$;
- E is a set of edges between elements, with $E \subseteq V \times V$;
- r is the root element of the rDAG.

A property of the rDAG deals with the **path**. In a rDAG, all elements can be reached from the root element. Given a $rDAG =< V, E, e_0 >$, $\forall$ element $e \in V$, $\exists$ a path $P(e_0, e) =< e_0, e_i, ..., e_j, e >$.

3 Quality of an Integrated Schema

The schema matching community lacks some metrics which evaluate the quality of an integrated schema. Indeed, some schema matching tools produce an integrated schema (with the set of mappings between input schemas). To the best of our knowledge, there are only a few metrics [4] for assessing the **quality of this integrated schema**. Namely, authors define two measures for integrated schema w.r.t. data sources. **Completeness** represents the percentage of concepts present in the data sources and which are covered by the integrated schema. **Minimality** checks that no redundant concept appears in the integrated schema. We have adapted these metrics for an expert integrated schema. Then, we complete them by another metric that evaluates the **structurality** of integrated schema. These three metrics are finally aggregated to evaluate the **schema proximity** of two schemas. To illustrate the schema proximity metric, we use the integrated schemas depicted by figures 2(a) and 2(b). Note that a set of mappings is necessarily provided with the integrated schema. Indeed, let us imagine that elements X and G match, i.e. they represent the same concept. This means that only one of them should be added in the integrated schema. On



(a) produced by a matching tool (b) given by an expert

Fig. 2. Two examples of integrated schemas

figure 2, we notice that X has been added in the tool's integrated schema while G appears in the expert integrated schema. Thus, with the set of mappings, we are able to check that the concept represented by X and G is present in the integrated schema, and only once.

3.1 Completeness and Minimality

In our context, we have an integrated schema produced by a matching tool, named Si_{tool}, and an expert integrated schema Si_{exp}. Recall that this expert integrated schema is ideal. $|Si_{exp}|$ stands for the number of elements in schema Si_{exp}. Thus, completeness, given by formula 1, represents the proportion of elements in the tool integrated schema which are common with the expert integrated schema. Minimality is computed thanks to formula 2, and it is the percentage of extra elements in the tool integrated schema w.r.t. expert integrated schema. Both metrics are in the range $[0, 1]$, with a 1 value meaning that the tool integrated schema is totally complete (respectively minimal) related to expert integrated schema.

$$comp(Si_{tool}, Si_{exp}) = \frac{|Si_{tool} \cap Si_{exp}|}{|Si_{exp}|} \tag{1}$$

$$min(Si_{tool}, Si_{exp}) = 1 - \frac{|Si_{tool}| - |Si_{tool} \cap Si_{exp}|}{|Si_{exp}|} \tag{2}$$

Let us compute completeness and minimality for the schemas shown in figure 2. As the number of common elements between the expert and tool integrated schemas is 6*, then completeness is equal to* $comp(Si_{tool}, Si_{exp}) = \frac{6}{7}$*. Indeed, we notice that the integrated schema produced by the matching tool lacks one element (G) according to the expert integrated schema. Similarly, we compute minimality, which gives us* $min(Si_{tool}, Si_{exp}) = 1 - \frac{8-6}{7} = \frac{5}{7}$*. The tool integrated schema is not minimal since two elements (X and Z) have been added w.r.t. the expert integrated schema.*

3.2 Structurality

Structurality denotes "*the qualities of the structure an object possesses*"[1]. To evaluate the structurality of a tool integrated schema w.r.t. an expert integrated schema, we check that each element owns the same ancestors.

The first step consists of converting the schemas into rooted directed acyclic graphs (DAG), which have been described in section 2. Consequently, integrated schemas Si_{exp} and Si_{tool} are respectively transformed into $rDAG_{exp}$ and $rDAG_{tool}$.

Secondly, for each element e_i from $rDAG_{exp}$ (except for the root), we build the two paths from the roots e_0 of both rDAGs. These paths are noted $P_{exp}(e_0, e_i)$ and $P_{tool}(e_0, e_i)$. We also remove from these paths element e_i. For sake of clarity, we respectively write P_{exp} and P_{tool} instead of $P_{exp}(e_0, e_i)$ and $P_{tool}(e_0, e_i)$. Note that if element e_i has not been included in $rDAG_{tool}$, then $P_{tool} = \emptyset$. From these two paths, we can compute the structurality of element e_i using formula 3. Intuition behind this formula is that element e_i in both integrated schemas shares the maximum number of common ancestors, and that no extra ancestor have been added in the tool integrated schema. Besides, an α parameter enables users to give a greater impact to the common ancestors to the detriment of extra ancestors. As the number of ancestors in P_{tool} might be large and cause a negative value, we constrain this measure to return a value between 0 and 1 thanks to a *max* function.

$$structElem(e_i) = \max\left(0, \frac{\alpha|P_{exp} \cap P_{tool}| - (|P_{tool}| - |P_{exp} \cap P_{tool}|)}{\alpha|P_{exp}|}\right) \quad (3)$$

Back to our example, we can compute the structurality of each (non-root) element from $rDAG_{exp}$, with a weight for α set to 2:

- ***B:*** *$P_{exp} = A$ and $P_{tool} = A$. Thus, $structElem(B) = \max(0, \frac{2\times1-(1-1)}{2\times1}) = 1$.*
- ***D:*** *$P_{exp} = A$ and $P_{tool} = A$. Thus, $structElem(D) = \max(0, \frac{2\times1-(1-1)}{2\times1}) = 1$.*
- ***E:*** *$P_{exp} = A, D$ and $P_{tool} = A, D$. Thus, $structElem(E) = \max(0, \frac{2\times2-(2-2)}{2\times2}) = 1$.*
- ***G:*** *$P_{exp} = A, D$ and $P_{tool} = \emptyset$. Thus, $structElem(G) = \max(0, \frac{2\times0-(0-0)}{2\times2}) = 0$.*
- ***C:*** *$P_{exp} = A, B$ and $P_{tool} = A, D$. Thus, $structElem(C) = \max(0, \frac{2\times1-(2-1)}{2\times2}) = \frac{1}{4}$.*
- ***F:*** *$P_{exp} = A, D$ and $P_{tool} = A$. Thus, $structElem(F) = \max(0, \frac{2\times1-(1-1)}{2\times2}) = \frac{1}{2}$.*

Finally, structurality of a tool integrated schema Si_{tool} w.r.t. an expert integrated schema Si_{exp} is given by formula 4. It is the sum of all element structuralities (except for the root element noted e_0) divided by this number of elements.

$$struct(Si_{tool}, Si_{exp}) = \frac{\sum_{i=1}^{i=n} structElem(e_i)}{n-1} \quad (4)$$

[1] http://en.wiktionary.org/wiki/structurality (March 2010)

In our example, structurality of the tool integrated schema is therefore the sum of all element structuralities. Thus, we obtain $struct(Si_{tool}, Si_{exp}) = \frac{1+1+1+0+\frac{1}{4}+\frac{1}{2}}{6} = 0.625$.

3.3 Integrated Schema Proximity

The integrated schema proximity, which computes the similarity between two integrated schemas, is a weighted average of previous measures, namely completeness, minimality and structurality. Three parameters (α, β and γ) enable users to give more weight to any of these measures. By default, these parameters are tuned to 1 so that the three measures have the same impact. Formula 5 shows how to compute schema proximity. It computes values in the range $[0, 1]$.

$$prox(Si_{tool}, Si_{exp}) = \frac{\alpha comp(Si_{tool}, Si_{exp}) + \beta min(Si_{tool}, Si_{exp}) + \gamma struct(Si_{tool}, Si_{exp})}{\alpha + \beta + \gamma} \quad (5)$$

In our example, the schema proximity between tool and expert integrated schemas is equal to $prox(Si_{tool}, Si_{exp}) = \frac{0.86+0.71+0.625}{3} = 0.73$ *with all parameters set to* 1*. Thus, the quality of the tool integrated schema is equal to* 73% *w.r.t. the expert integrated schema.*

3.4 Discussion

We now discuss some issues dealing with the proposed schema proximity metric.

Contrary to [6], our structurality metric does not rely on discovering common subtrees. We mainly check for common ancestors for each element and do not penalise some elements. For instance, child elements whose parent element is different are not included in a subtree, and they are taken in account as single elements (not part of a subtree) when measuring the schema quality . With our structurality metric, we avoid this problem since each element with its ancestors is individually checked.

We have decided to exclude the root element from the metric, because it already has a strong weight due to its position. If the root element of the tool integrated schema is the same than the one in the expert integrated schema, then all elements (present in both schemas) which are compared already have a common element (the root). Conversely, if the root elements of both integrated schemas are different, then comparing all elements involves a decreased structurality due to the different root elements. Therefore, there was no need to consider this root element.

Our measure assumes that a set of mappings between the source schemas has been discovered. This set of mappings has a strong impact for building the integrated schema. In most cases, domain experts can check and validate the mappings, so that mapping errors do not affect the quality of the integrated schema. However, there also exists many cases in which manual checking is not possible, e.g., in dynamic environments or in large scale scenarios. What is the influence of mapping quality in such contexts ? Let us discuss these points according to precision and recall. The former measure denotes the percentage of correct mappings among all those which have been discovered. In other words, the lowest the precision is, the more incorrect mappings the tool has discovered. For all these incorrect mappings, only one element of the mapping is chosen to be included in the integrated schema, while the other is not. Since the mapping is incorrect, all elements composing it should have been put in the schema. Thus, precision has an influence on completeness. On the contrary, the second measure, recall, directly impacts minimality. As it computes the percentageof correct mappings that have been discovered among all correct mappings, it evaluates the number of correct mappings that have been "missed" by the tool. A "missed" mapping is fully integrated, i.e., all of its elements are added in the integrated schema. Yet, only one of them should be added. For these reasons, the quality of the set of mappings is strongly correlated with the quality of the integrated schema.

Although one could see the requirement of an expert integrated schema as a drawback, we advocate that the measures to evaluate the quality of mappings (precision and recall) are also based on the existence of an expert ground truth. Besides, authors of [2] indicate that companies and organizations often own global repositories of schemas or common vocabularies. These databases can be seen as incomplete expert integrated schemas. Indeed, they have mainly been manually built, thus ensuring an acceptable quality. They are also incomplete since all schemas or all of their underlying concepts are not integrated in these databases. Yet, it is possible to use them as ground truth. Let us imagine that users of a company are accustomed to a global repository. If the company needs an extended integrated schema which includes the concepts of the global repository, it could be convenient for the users that the new integrated schema keeps a similar structure and completeness with the one of the global repository. In this case, we can apply our measures both on the global repository and the new integrated schema to check if these constraints are respected.

However, the integrated schema proximity metric does not take into account user requirements and other constraints. For instance, a user might not want a complete integrated schema since (s)he will query only a subset of the schema. Or the minimality could not be respected because the application domain requires some redundancies. In another way, some hardware constraints may also impact integrated schemas.

4 Experiments Report

In this section, we present the evaluation results of the following schema matching tools: COMA++ [7,8] and Similarity Flooding (SF) [9,10]. These tools are described in the next section (see section 5). We notice that it is hard to find available schema matchers to evaluate. We first describe our experiment protocol, mainly the datasets that we used. We then report results achieved by schema matching tools on the quality of integrated schema by datasets. Due to space limitation, we do not include quality (in terms of precision, recall or F-measure) obtained by the tools for discovering the mappings. As explained in Section 3.4, mapping discovery is a crucial initial step for building the integrated schema. Thus, we provide some figures when necessary to justify the quality of the integrated schema.

4.1 Experiments Protocol

Here are the datasets used for these experiments:

- **Person dataset.** contains two small-sized schemas describing a person. These schemas are synthetic.
- **Order dataset.** deals with business. The first schema is drawn from the XCBL collection[2], and it owns about 850 elements. The second schema (from OAGI collection[3]) also describes an order but it is smaller with only 20 elements. This dataset reflects a real-case scenario in which a repository of schemas exist (similar to our large schema) and the users would like to know if a new schema (the small one in our case) is necessary or if a schema or subset of a schema can be reused from the repository.
- **University courses dataset.** These 40 schemas have been taken from Thalia collection presented in [11]. Each schema has about 20 nodes and they describe the courses offered by some worldwide universities. As described in [2], this datasets could refer to a scenario where users need to generate an exchange schema between various data sources.
- **Biology dataset.** The two large schemas come from different collections which are protein domain oriented, namely Uniprot[4] and GeneCards [5]. This is an interesting dataset for deriving a common specific vocabulary from different data sources which have been designed by human experts.
- **Currency** and **sms** datasets are popular web services which can be found at *http://www.seekda.com*
- **University department dataset.** describes university departments and it has been widely used in the literature [12]. These two small schemas have very heterogeneous labels.
- **Betting.** contains tens of webforms, extracted from various websites by the authors of [13]. As explained by authors of [2], schema matching is often a process which evaluates the costs (in terms of resources and money) of a project, thus indicating its feasibility. Our *betting* dataset can be a basis for project planning, i.e., to help users decide if integrating their data sources is worth or not.

4.2 Experiments

For each schema matching tool, we have first run the schema matching process to discover mappings between source schemas. Thanks to these mappings (which have not been manually checked), the tools have then built an integrated schema. All experiments were run on a 3.0 Ghz laptop with 2G RAM under Ubuntu Hardy.

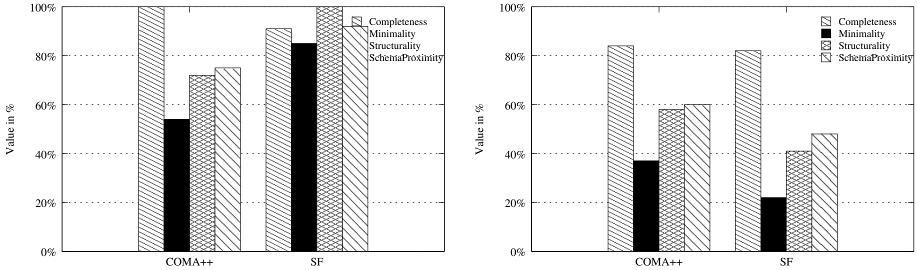
[2] www.xcbl.org

[3] www.oagi.org

[4] http://www.ebi.uniprot.org/support/docs/uniprot.xsd

[5] http://www.geneontology.org/GO.downloads.ontology.shtml

Betting dataset. Figure 3(a) depicts the quality for the *betting* dataset. COMA++ successfully encompasses all concepts (100% completeness) while SF produces the same structure than the expert (100% structurality). Both tools did not achieve a minimal integrated schema, i.e., without redundancies. SF generates the most similar integrated schema w.r.t. the expert one (schema proximity equal to 92%).

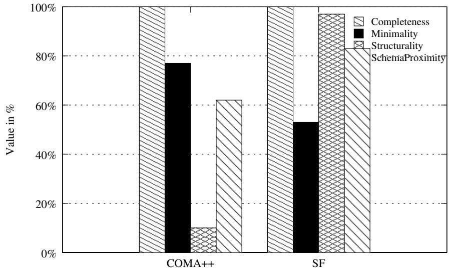


(a) Quality for the *betting* dataset (b) Quality for the *biology* dataset

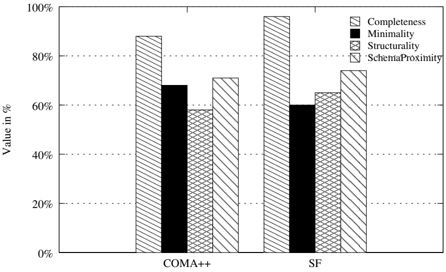
Biology dataset. With this large scale and domain-specific dataset, the schema matching tools have poorly performed for discovering mappings (less than 10% F-measure). These mitigated results might be explained by the fact that no external resource (e.g., a domain ontology) was provided. However, as shown by figure 3(b), the tools were able to build integrated schemas with acceptable completeness (superior to 80%) but many redundancies (minimality inferior to 40%) and different structures (58% and 41% structuralities). These scores can be explained by the failure for discovering correct mappings. As a consequence, lots of schema elements have been added into the integrated schemas, including redundant elements. For structurality, we believe that for unmatched elements, the schema matching tools have copied the same structure than the one of the input schemas.

Currency dataset. On figure 3(c), we can observe the quality of the integrated schemas built by COMA++ and SF for *currency*, a nested average-sized dataset. This last tool manages to build a more similar integrated schema (83% schema proximity against 62% for COMA++). Although both tools have a 100% completeness, COMA++ avoids more redundancies (due to a better recall during mapping discovery) while SF respects more the schema structure. We notice that COMA++ produces a schema with a different structure than the one of the expert. This is probably due to

Order dataset. This experiment deals with large schemas whose labels are normalised. Similarly to the other large scale scenario, schema matching tools do not perform well for this *order* dataset (F-measures less than 30%). As for quality of the integrated schema, given by figure 3(d), both tools achieve a schema proximity above 70%, with a high completeness.



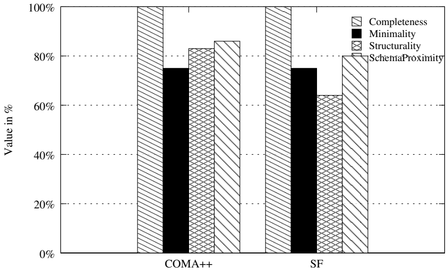
(c) Quality for the *currency* dataset



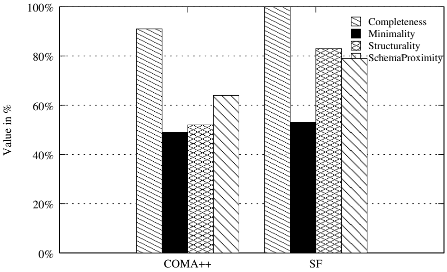
(d) Quality for the *order* dataset

Person dataset. Figure 3(e) depicts quality for the *person* dataset, which contains small schemas featuring low heterogeneity in their labels. We notice that both generated schemas are complete and they achieve the same minimality (76%). However, for this dataset containing nested schemas, COMA++ is able to respect a closer structurality than SF. The tools achieve a 80% schema proximity, mainly due to the good precision and recall that they both achieve.

Sms dataset. The *sms* dataset does not feature any specific criteria, but it is a web service. A low quality for discovering mappings has been achieved (all F-measures below 30%). As they missed many correct mappings, the integrated schemas produced by the tools have a minimality around 50%, as shown on figure 3(f). SF obtains better completeness and structurality than COMA++.



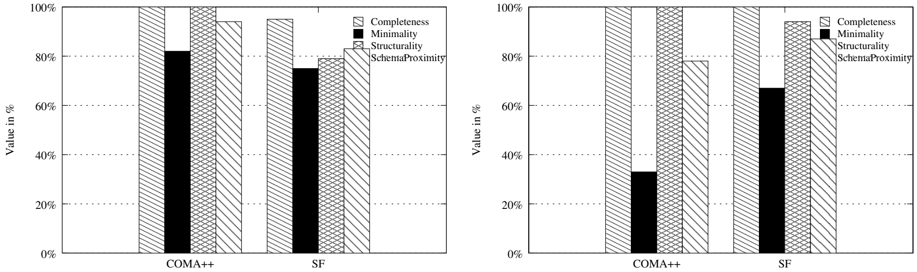
(e) Quality for the *person* dataset



(f) Quality for the *sms* dataset

Univ-courses dataset. The *univ-courses* dataset contains flat and average-sized schemas. On figure 3(g), the quality of COMA++ and SF's integrated schemas are evaluated. It appears that both tools produces an acceptable integrated schema w.r.t. the expert one (schema proximity equal to 94% for COMA++ and 83% for SF). Notably, COMA++ achieves a 100% completeness and 100% structurality.

Univ-dept dataset. The last dataset, *univ-dept*, has been widely used in the litterature. It provides small schemas with high heterogeneity and the results of the schema matching tools are shown on figure 3(h). Both tools achieve acceptable completeness and structurality (all above 90%), but they have more difficulties to respect the minimality constraint, merely due to their average recall.



(g) Quality for the *univ-courses* dataset (h) Quality for the *univ-dept* dataset

4.3 Concluding the Experiments Report

We conclude this section by underlining some general points about these experiments.

- Average completeness (for all tools and all datasets) is equal to 91%. On the contrary, average minimality is 58% and average structurality reaches 68%. Indeed, schema matching tools mainly promote precision, thus they avoid the discovery of incorrect mappings and they do not miss too many schema elements when building the integrated schema. A lower recall means that many similar schema elements are added in the integrated schema, thus reducing minimality.
- We also notice that it is possible to obtain a high minimality with a low recall, if precision is low too. Indeed, the low recall means that we have missed many correct mappings, thus two similar elements could be added twice in the integrated schema. But with a low precision, there are many incorrect discovered mappings, and only one of their elements would be added in the integrated schema. As an example, let us imagine that a correct mapping between elements *A* and *A'* is not discovered. Both *A* and *A'* are added in the integrated schema, unless one of them has been incorrectly matched to another element. This explains the high minimality achieved with some datasets, despite of a low recall.
- Similarity Flooding provides a better quality when building integrated schemas (79% average schema proximity against 67% for COMA++).
- If a correct mapping is missed by a matching tool, then both elements of this missed mapping are added in the integrated schema. Structurality only takes into account one of these elements (the one which is in the expert integrated schema). The other is ignored, but it also penalizes minimality. This explains why structurality and completeness have high values even when mapping quality measures return low values.
- Schema proximity is also quite high, simply because it averages completeness and structurality values which are already high. For instance, when a few correct mappings are discovered (*order* or *biology* datasets), many elements are added into integrated schema, thus ensuring a high completeness but a low minimality. Due to the missed mappings, lots of elements have to be added into the integrated schema, and the easiest way is to keep the

same structure that can be found in the source schemas, thus guaranting an acceptable structurality. However, our schema proximity measure can be tuned (with parameters α, β and γ) to highlight a weakness in any of the three criteria (completeness, minimality or structurality).

5 Related Work

Many approaches have been devoted to schema matching. In [14,15], authors have proposed a classification for matching tools, which has been later refined in [16]. Similarly, ontology researchers are also prolific for designing approaches to fulfill the alignment task between ontologies [17]. However, the yearly OAEI challenge[6] for instance mainly evaluates the mapping quality, and not ontology integration. This section only focuses on schema matching tools which are publicly available for evaluation with our benchmark, namely Similarity Flooding and COMA++.

5.1 Similarity Flooding/Rondo

Similarity Flooding [9] (also called Rondo [10]) is a neighbour affinity matching tool. First, it applies a terminological similarity measure to discover initial correspondences, and then feeds them to the structural matcher for propagation. The weight of similarity values between two elements is increased, if the algorithm finds some similarity between related elements of the pair. The user can then (in)validate the discovered correspondences, and the tool builds an integrated schema based on these correspondences.

5.2 COMA/COMA++

COMA/COMA++ [18,7] is a generic, composite matcher with very effective matching results. The similarity of pairs of elements is calculated using linguistic and terminological measures. Then, a strategy is applied to determine the pairs that are presented as correspondences. COMA++ supports a number of other features like merging, saving and aggregating match results of two schemas.

6 Conclusion

In this paper, we have presented new measures for assessing the quality of integrated schema produced by schema matching tools. Namely, we are able to evaluate the structure of this schema. Combined with minimality and completeness, the schema proximity measure computes the likeness of an integrated schema w.r.t. an expert one. We have finally evaluated two schema matching tools, COMA++ and Similarity Flooding, over 10 datasets. The resulting report indicates that Similarity Flooding generates better integrated schemas. But it also shows that schema matching tools could be enhanced to let users express some constraints for generating an integrated schema, for instance in terms of design.

[6] http://oaei.ontologymatching.org/

As future work, we intend to enhance our measures for ontologies. The structurality measure should be refined to express the different relationships (e.g., generalization, instance) between the paths of two elements. As many organizations own schema repositories which could be used as expert integrated schemas, we also plan to extend our measures for reflecting the incompleteness of these repositories.

References

1. Batini, C., Lenzerini, M., Navathe, S.B.: A Comparitive Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys 18(4), 323–364 (1986)
2. 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)
3. Castano, S., De Antonellis, V., Fugini, M.G., Pernici, B.: Conceptual schema analysis: techniques and applications. ACM Trans. Database Syst. 23(3), 286–333 (1998)
4. da Conceição Moraes Batista, M., Salgado, A.C.: Information quality measurement in data integration schemas. In: QDB, pp. 61–72 (2007)
5. Kesh, S.: Evaluating the quality of entity relationship models. Information and Software Technology 37, 681–689 (1995)
6. Duchateau, F., Bellahsene, Z., Hunt, E.: Xbenchmatch: a benchmark for xml schema matching tools. In: VLDB, pp. 1318–1321 (2007)
7. Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++. In: ACM SIGMOD Conference, DEMO paper, pp. 906–908 (2005)
8. Do, H.H., Rahm, E.: Matching large schemas: Approaches and evaluation. Information Systems 32(6), 857–885 (2007)
9. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: ICDE, pp. 117–128 (2002)
10. Melnik, S., Rahm, E., Bernstein, P.A.: Developing metadata-intensive applications with rondo. J. of Web Semantics I, 47–74 (2003)
11. Hammer, J., Stonebraker, M., Topsakal, O.: Thalia: Test harness for the assessment of legacy information integration approaches. In: Proceedings of ICDE, pp. 485–486 (2005)
12. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Ontology matching: A machine learning approach. In: Handbook on Ontologies in Information Systems (2004)
13. Marie, A., Gal, A.: Boosting schema matchers. In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part II. LNCS, vol. 5332, pp. 283–300. Springer, Heidelberg (2008)
14. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal 10(4), 334–350 (2001)
15. Euzenat, J., et al.: State of the art on ontology matching. Technical Report KWEB/2004/D2.2.3/v1.2, Knowledge Web (2004)
16. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. Journal of Data Semantics IV, 146–171 (2005)
17. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, Heidelberg (2007)
18. Do, H.H., Rahm, E.: COMA - A System for Flexible Combination of Schema Matching Approaches. In: VLDB, pp. 610–621 (2002)