

A Survey and Categorization of Ontology-Matching Cases

Zharko Aleksovski^{1,2}, Willem Robert van Hage^{2,3}, and Antoine Isaac²

¹ Philips Research, Eindhoven,

² Vrije Universiteit, Amsterdam

³ TNO Science & Industry, Delft

{zharko,wrvhage,aisaac}@few.vu.nl

Abstract. Methodologies to find and evaluate solutions for ontology matching should be centered on the practical problems to be solved. In this paper we look at matching from the perspective of a practitioner in search of matching techniques or tools. We survey actual matching use cases, and derive general categories from these. We then discuss the value of existing techniques for these categories.

1 Introduction

For an application developer to know which ontology matching system best suits his needs, application requirements have to be taken into account. Recently, innovative work carried out in the KnowledgeWeb network of excellence [1] has analyzed the requirements of usage scenarios and proposed a case-based recommendation method: a given application is profiled along different dimensions – input, usage, etc. This profile is then compared to a characterization of different matching tools, to determine which tool best fits the case that is considered. This method is based on a characterization of existing matching solutions obtained by the carefully crafted benchmark test of the Ontology Alignment Evaluation Initiative⁴ (OAEI). However, the focus of the OAEI has been mainly on comparing techniques for research. As a result, the categories that are used are not straightforwardly linked to real-world cases.

To give answers for application developers it is necessary to build better links between application specifications and matching systems. A possible way to do so is to position each new matching case with respect to an matching-oriented categorization of applications built from the characterization of these cases in terms of matching requirements and performances of different techniques. [1] actually initiates such an effort: the authors gather use cases, point at the typical operations (data transformation, ontology merging, etc.) and elicit some matching quality requirements for them (correctness, completeness). Yet, the cases considered there are abstract scenarios. In this paper, we intend to provide a better application grounding for case-based recommendation by turning to lessons learned from concrete cases. Our contribution consists in a categorization-oriented survey of existing ontology matching cases trying to give answers to the following questions: (i) what are the different kinds of cases in which

⁴ <http://oaei.ontologymatching.org>

ontology matching has been deployed so far? (ii) Can we observe common aspects, leading to a classification of these cases? (iii) Are there matching techniques that have been observed to perform better on specific sets of cases?

We first analyse existing documented ontology-matching use cases, highlighting their main requirements (Section 2). We notice the emergence of four different categories of use cases, depending on their purpose, the data they deal with, and their priorities regarding matching qualities: data migration, question answering, serendipity in browsing, and unified view over collections (Section 3). Section 4 extends these considerations towards the realm of ontology-matching tools, by showing how specific matching techniques perform better for specific classes. Section 5 concludes the paper.

2 Ontology-Matching Cases

The importance of ontology matching was identified through various scenarios which require its solution. Such scenarios, for example described in [2], are: agent communication, emergent semantics, P2P information sharing, personal information delivery, etc. However, these stand for *possible* uses of matching. To carry out the analysis grounding our case categorization and recommendations for matching techniques, we have instead investigated examples of reported matching applications. Such cases had to: (i) provide information on the actual alignment, (ii) report on which techniques can be used to solve the matching problem, and (iii) clearly describe how the correspondences will be used in an application. As a consequence we did not take examples from real-time matching cases like negotiation, where the mapping data is usually generated dynamically, or cases such as the one reported in [3] where the application of the correspondences is not explicitly defined.

The ten cases we have selected are: *MACS*, *Agricultural Thesauri*, *Renardus*, *STITCH browser*, *WebDewey*, *Intensive Care*, *the High Performance Knowledge Base*, *the Unified Medical Language System*, *Internet Music Schemas* and *Internet Directories*. We detail here only one example, and refer the reader to the companion webpage, <http://www.few.vu.nl/~aisaac/iswc2007/cases>, for the other descriptions.

Intensive Care In this use case the alignment is needed for data-migration purposes. The alignment is directly applied for classification reasoning. Two Amsterdam hospitals, OLVG and AMC, own controlled unstructured vocabularies for registering reasons for patient admission to the intensive care units. The vocabularies are lists of terms and every time a patient enters intensive care she is assigned one of these terms. Correspondences between OLVG and AMC classes are required in order to migrate the patient data from OLVG to the AMC vocabulary. The OLVG vocabulary contains 1,399 terms and AMC 1,460 [4]. To test the performance of automatic matching techniques, a gold standard was created by a medical expert for a sample of 200 OLVG classes. For 37% of OLVG terms in the corpus the expert found no correspondences, for 36% he found correspondences with large lexical similarity between the corresponding terms, and for the remaining 37% he found correspondences with no lexical overlap, which would require the use of some kind of background knowledge. An example of lexical correspondence is Brain tumor to Braintumor, and example of a correspondence that requires background knowledge is Heroin intoxication to Drugs overdose.

3 Descriptions of Problem Types

Precision versus Recall Three tasks determine the time cost of applying ontology matching: (i) preparation and actual alignment, (ii) assessing correspondences and (iii) adding missing correspondences. The goal of an automatic system is to reduce the amount of time a user spends on these tasks. Usually, better performance for one task means worse performance for another. Good representation standards and a fast system optimize on the first task, a system with high Precision optimizes on the second task, and a system with high Recall optimizes on the third task. For some cases assessing a single correspondence is time consuming (*e.g.* if concepts are imprecisely defined). For others it is very time consuming to find a missing correspondence (*e.g.* if ontologies are huge). Hence, each case has its own optimal combination of Precision and Recall.

Complexity of representation Each use case requires a different level of knowledge-representation complexity. Some use cases only need basic semantic structures, others need rich ontologies with many different properties and use of logical axioms.

Four categories of use-cases Some of the use cases we presented use the alignment for a similar purpose. Some use the alignment primarily to enrich the descriptions of the data, while others use them primarily to enlarge the data collections. Further, we have noticed that use cases with the same problem type often have similar performance requirements for the ontology matching and use a similar level of knowledge-representation complexity. We propose to categorize the use cases according to the following four types of problems.

Question answering. This problem type is characterized by an emphasis on precise results and the need for highly complex knowledge sources. The use cases we found aim at providing detailed factual information about the data. For example, “*What is the connection between trombose and mortality?*” or “*Who was the president of the United States of America in 1965?*”. As opposed to the real-time question answering described in [1] the goal of these use cases is not to provide a complete list of answers, but one very precise answer (*cf.* [5]). Use cases of this type are the *High-Performance Knowledge Base* case and (partly) the *Unified Medical Language System* case.

Unified view over collections. This problem type is characterized by a balanced need for Precision and Recall and knowledge sources of medium complexity. Examples of such sources include traditional thesauri and thesauri with added relations, such as artist-style links, part-whole, or tool-action. The use cases we found aim at providing unified access to heterogeneous collections that are usually maintained by different authorities. Use cases of this type are the *STITCH browser* case, *MACS*, *Renardus*, the *Agricultural Thesauri* case and, to a lesser extent, *WebDewey*.

Serendipity in browsing. This problem type is characterized by an emphasis on Recall and relatively simple knowledge structures, such as taxonomies. The use cases we found aim at joining taxonomies to enlarge the collection, to provide users with instances they did not know before. Use cases of this type are the *Internet Music Schemes* and the *Internet Directories* cases.

Data migration. Like Question Answering, this problem type emphasizes the need for precise results, not necessarily requiring complex knowledge sources. The use case of this type that we found, the *Intensive Care* case, aims at re-classifying existing instances into classes from a newer schema.

4 Techniques that solve the Mapping Problem in the Use Cases

Observing the cases leaves an open question: which techniques can actually produce the required correspondences? Here, we consider four different types of matching techniques: lexical – based on lexical comparisons of labels and glosses, structural – using the structure of the ontologies, background knowledge – using additional external knowledge, and instance-based – using classified instance data.

Question answering. In these cases the ontologies are usually vast and complex. They have substantial lexical overlap, but the use of different naming conventions and of the same names in different contexts prevents straightforward lexical detection of the correspondences. As reported in the UMLS and HPKB examples, the first problem can be overcome by detecting the patterns used in naming, and then normalizing the names so that lexical techniques can find the correspondences. The second problem, detecting the context, can be solved by taking into account the domain of the ontologies and using their structure. In UMLS, for instance, if a concept Kidney is found in a classification of diseases or is a subconcept of concept Diseases, then it surely refers to problems related to kidneys. Practical cases indicate thus that lexical and structural techniques are good candidate solutions for question answering matching use cases.

Unified view over collections. Here, naming conventions and modeling decisions may differ, but lexical matching solves a large part of the problem. The vocabularies to align can have quite a broad coverage or shared domain and concerns. Often – *e.g.*, when jargon differs – background knowledge is however needed. If different languages are used it even becomes crucial, either in the form of a multilingual “rosetta stone” or of a translation service. Structure-based techniques are generally of much less use: *e.g.*, the semantic link that come in the thesauri can be used for meaning disambiguation, but this makes them a secondary source for matching information, not reported to contribute significantly in the examples. Furthermore, as the principles and coherence of the structure can vary from one thesauri to another, these techniques might prove unreliable. Finally, the instance data found in several collections can prove very useful, as the meaning of manipulated concepts is assumed to be ultimately given by the items that are categorized with their help.

Serendipity in browsing. Here, lexical methods are reported to perform poorly, which is caused by two problems: ambiguity in naming concepts and lack of standardized criteria for classifying instance data. Among the artists shared by two portals of the music case, only 38% of the ones classified in the Rock genre in one dataset fall in the Rock category defined by the other. The first problem can be approached by considering the context in which the concepts appear. The second problem can only be solved by matching the instance data. Actually, when matching based on instance data, one has to consider the instances in the subclasses of a given class. As reported in the Internet Directory case, this makes big difference in the performance of the alignment.

Data migration. In the *Intensive care* case, the vocabularies have no structure. Furthermore, there is no substantial, explicitly shared instance data, since the goal is to transfer the instance data itself from one system to the other. This leaves two options for a solution: lexical techniques and background knowledge. These have actually turned out to be sufficient to solve the problem in this specific example.

5 Conclusions and Future Work

Focusing on how to find a good matching method for a given application, we have surveyed a number of real-world ontology matching use cases and proposed a categorization of them in four groups, based on the applicative purpose of the alignment. We have then positioned each category in way compatible with a principled (benchmark-based) profiling of different matching techniques. These can then be selected based on their matching the criteria coming with a given application.

More descriptions of realistic use cases (where the alignment is applied in practice) are clearly needed to complement the analysis presented here, especially to get a better coverage of new innovative scenarios still being investigated now, like semantic web agent communication. It would also be interesting to investigate some cases coming from the database domain, as our survey is quite biased towards alignment cases for description *vocabularies* – as opposed to description *structures*. Such accounts exist [6, 7] but the database research community, as the semantic web one, seems to have put more effort on describing tools and methods than cases [8, 9].

A particular emphasis shall be put on revealing application-specific limitations of matching techniques, as when dealing with specific naming schemes or underspecified structural links. Better consideration of such application-specific constraints is necessary for future benchmarking efforts. This way, the ontology-matching research community could also fully benefit from the surveying effort.

Acknowledgments

We would like to thank Frank van Harmelen, Guus Schreiber, Lourens van der Meij, Stefan Slobach, Shenghui Wang (VU), Hap Kolb (TNO), Merlijn Sevenster, Warner ten Kate (Philips Research), Margherita Sini (FAO), Lori Finch (NAL). Our work is partly funded by NWO, the Netherlands Organisation for Scientific Research (STITCH project) and the Virtual Laboratories for e-Science (VL-e) project.

References

1. Euzenat, J., Ehrig, M., Jentzsch, A., Mochol, M., Shvaiko, P.: Case-based recommendation of matching tools and techniques. KnowledgeWeb Project deliverable D1.2.2.2.1 (2007)
2. Euzenat, C.J.: D2.2.3: State of the art on ontology alignment (2004)
3. Zhang, S., Bodenreider, O.: Aligning representations of anatomy using lexical and structural methods. AMIA Symposium (2003) 753–757
4. Aleksovski, Z., Klein, M., ten Kate, W., van Harmelen, F.: Matching unstructured vocabularies using a background ontology. In: Proc. of EKAW. (2006)
5. Voorhees, E., Tice, D.: The TREC-8 question answering track evaluation. In: Proc. of TREC-8. (1999)
6. Clifton, C., Housman, E., Rosenthal, A.: Experience with a combined approach to attribute-matching across heterogeneous databases. In: IFIP Conference on Data Semantics. (1997)
7. Rosenthal, A., Seligman, L., Renner, S.: From semantic integration to semantics management: case studies and a way forward. Sigmod Records Special Section **33**(4) (2004)
8. Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. VLDB Journal **10**(4) (2001)
9. Doan, A., Halevy, A., Renner, S.: Semantic integration research in the database community: A brief survey. AI Magazine **26**(1) (2005)