Ontology Matching

OM-2009

Papers from the ISWC Workshop

Introduction

Ontology matching is a key interoperability enabler for the Semantic Web, as well as a useful tactic in some classical data integration tasks. It takes the ontologies as input and determines as output an alignment, that is, a set of correspondences between the semantically related entities of those ontologies. These correspondences can be used for various tasks, such as ontology merging and data translation. Thus, matching ontologies enables the knowledge and data expressed in the matched ontologies to interoperate.

The workshop has three goals:

- To bring together leaders from *academia*, *industry* and *user institutions* to assess how academic advances are addressing real-world requirements. The workshop will strive to improve academic awareness of industrial and final user needs, and therefore, direct research towards those needs. Simultaneously, the workshop will serve to inform industry and user representatives about existing research efforts that may meet their requirements. The workshop will also investigate how the ontology matching technology is going to evolve.

- To conduct an extensive and rigorous evaluation of ontology matching approaches through the OAEI (Ontology Alignment Evaluation Initiative) 2009 campaign, `http://oaei.ontologymatching.org/2009`. This year's OAEI campaign introduces two new tracks about oriented alignments and about instance matching (a timely topic for the linked data community). Therefore, the ontology matching evaluation initiative itself will provide a solid ground for discussion of how well the current approaches are meeting business needs.

- To examine similarities and differences from database schema matching, which has received decades of attention but is just beginning to transition to mainstream tools.

We received 25 submissions for the technical track of the workshop. The program committee selected 6 submissions for oral presentation and 12 submissions for poster presentation. 16 matching systems participated in this year's OAEI campaign. Further information about the Ontology Matching workshop can be found at: `http://om2009.ontologymatching.org/`.

Acknowledgments. We thank all members of the program committee, authors and local organizers for their efforts. We appreciate support from the Trentino as a Lab (TasLab)[1] initiative of the European Network of the Living Labs[2] at Informatica Trentina SpA[3] and the EU SEALS (Semantic Evaluation at Large Scale)[4] project.

Pavel Shvaiko
Jérôme Euzenat
Fausto Giunchiglia
Heiner Stuckenschmidt
Natasha Noy
Arnon Rosenthal

October 2009

[1] http://www.taslab.eu
[2] http://www.openlivinglabs.eu
[3] http://www.infotn.it
[4] http://www.seals-project.eu

Organization

Organizing Committee

Pavel Shvaiko, TasLab, Informatica Trentina SpA, Italy
Jérôme Euzenat, INRIA & LIG, France
Fausto Giunchiglia, University of Trento, Italy
Heiner Stuckenschmidt, University of Mannheim, Germany
Natasha Noy, Stanford Center for Biomedical Informatics Research, USA
Arnon Rosenthal, The MITRE Corporation, USA

Program Committee

Yuan An, Drexel University, USA
Zohra Bellahsene, LIRMM, France
Paolo Besana, University of Edinburgh, UK
Olivier Bodenreider, National Library of Medicine, USA
Isabel Cruz, University of Illinois at Chicago, USA
Jérôme David, Université Pierre Mendes-France, INRIA & LIG, France
Avigdor Gal, Technion, Israel
Jingshan Huang, University of South Alabama, USA
Wei Hu, Southeast University, China
Ryutaro Ichise, National Institute of Informatics, Japan
Antoine Isaac, Vrije Universiteit Amsterdam, Netherlands
Krzysztof Janowicz, University of Muenster, Germany
Chiara Ghidini, Fondazione Bruno Kessler (IRST), Italy
Bin He, IBM, USA
Yannis Kalfoglou, Ricoh Europe plc., UK
Monika Lanzenberger, Vienna University of Technology, Austria
Patrick Lambrix, Linköpings Universitet, Sweden
Maurizio Lenzerini, University of Rome - Sapienza, Italy
Vincenzo Maltese, University of Trento, Italy
Fiona McNeill, University of Edinburgh, UK
Christian Meilicke, University of Mannheim, Germany
Luca Mion, TasLab, Informatica Trentina SpA, Italy
Peter Mork, The MITRE Corporation, USA
Leo Obrst, The MITRE Corporation, USA
Massimo Paolucci, DoCoMo Labs, Germany
François Scharffe, INRIA & LIG, France
Umberto Straccia, ISTI-C.N.R., Italy
York Sure, University of Koblenz, Germany
Andrei Tamilin, Fondazione Bruno Kessler (IRST), Italy
Lorenzino Vaccari, PAT, Italy

Ludger van Elst, DFKI, Germany
Frank van Harmelen, Vrije Universiteit Amsterdam, Netherlands
Yannis Velegrakis, University of Trento, Italy
Baoshi Yan, Bosch Research, USA
Rui Zhang, University of Trento, Italy
Songmao Zhang, Chinese Academy of Sciences, China

Additional Reviewers

Fabien Duchateau, LIRMM, France
Christophe Guéret, Vrije Universiteit Amsterdam, Netherlands
Qiang Liu, Linköpings Universitet, Sweden
Shenghui Wang, Vrije Universiteit Amsterdam, Netherlands

Table of Contents

PART 1 - Technical Papers

PART 2 - OAEI Papers

PART 3 - Posters

Scalable Matching of Industry Models – A Case Study

Brian Byrne[1], Achille Fokoue[2], Aditya Kalyanpur[2], Kavitha Srinivas[2], and Min Wang[2]

[1] IBM Software Group, Information Management, Austin, Texas
byrneb@us.ibm.com
[2] IBM T. J. Watson Research Center, Hawthorne, New York
achille, adityakal, ksrinivs, min@us.ibm.com

Abstract. A recent approach to the problem of ontology matching has been to convert the problem of ontology matching to information retrieval. We explore the utility of this approach in matching model elements of real UML, ER, EMF and XML-Schema models, where the semantics of the models are less precisely defined. We validate this approach with domain experts for industry models drawn from very different domains (healthcare, insurance, and banking). We also observe that in the field, manually constructed mappings for such large industry models are prone to serious errors. We describe a novel tool we developed to detect suspicious mappings to quickly isolate these errors.
keywords: Model matching.

1 Introduction

The world of business is centered around information. Every business deals with a myriad of different semantic expressions of key business information, and expends huge resources working around the inconsistencies, challenges and errors introduced by a variety of information models. Typically, these information models organize the data, services, business processes, or vocabulary of an enterprise, and they may exist in different forms such as ER models, UML models, thesauri, ontologies or XML schema. A common problem is that these varying models rarely share a common terminology, because they have emerged as a result of several inputs. In some cases, mergers of organizations operating in the same business result in different information models, to express the same exact concepts. In other cases, they may have been developed by different organizational units to express overlapping business concepts, but in slightly different domains.

Irrespective of how these models came about, today's business is faced with many different information models, and an increasing need to integrate across these models, through data integration, shared processes and rules, or reusable business services. In all of these cases, the ability to relate, or map, between different models is critical. Both human attempts to manually map different information models and and the use of tools to automate mappings however are very error prone in the real world. For humans, the source of the error comes from multiple sources:

– The size of these models (typically, these models have several thousand elements each)
– The fact that lexical names of model elements rarely match, or when they do match, its because of the wrong reasons (e.g., a document may have an endDate attribute, as does a claim, but the two endDate reflect semantically different things, although they match at the lexical level).
– Models often express concepts at different levels of granularity, and it may not always be apparent at what level the concept should be mapped. In many real world mappings, we have observed a tendency for human analysts to map everything to generic concepts rather than more specific concepts. While these mappings are not necessarily invalid, they have limited utility in data integration scenarios, or in solution building.

The above points make it clear that there is a need for a tool to perform semi-automated model mapping, where a tool can help suggest appropriate mappings to a human analyst. Literature on ontology matching and alignment is clearly helpful in designing such a tool. Our approach to building such a tool is similar in spirit to the ideas implemented in Falcon-AO [1],[2] and PRIOR ([3]), except that we adapted their techniques to UML, ER and EMF models. Matching or alignment across these models is different from matching ontologies, because the semantics of these models are poorly defined compared to those of ontologies. Perhaps due to this reason, schema mapping approaches tend to focus mostly on lexical and structural analysis. However, existing schema mapping approaches scale very poorly to large models. Most analysts in the field therefore tend to revert to manual mapping, despite the availability of many schema mapping tools.

We however make the observation that in most industry models, the semantics of model elements is buried in documentation (either within the model, or in separate PDF, Excel or Word files). We therefore use techniques described by Falcon-AO and PRIOR to build a generic representation that allows us to exploit the structural and lexical information about model elements along with semantics in documentation. The basic idea, as described in PRIOR is to convert the model mapping problem into a problem of information retrieval. Specifically, each model element is converted into a virtual document with a number of fields that encode the structural, lexical and semantic information associated with that model element. This information is in turn expressed as a term vector for a document. Mapping across model elements is then measured as a function of document similarity; i.e., the cosine similarity between two term vectors. This approach scales very well because we use the Apache Lucene text search engine for indexing and searching these virtual documents.

The novelty in our approach is that we also developed an engine to identify suspicious mappings produced either by our tool or by human analysts. We call this tool a Lint engine for model mappings, after the popular Lint tool which checks C programs for common software errors. The key observation that motivated our development of the Lint engine was that human model mappings

were shockingly poor for 3/4 model mappings that were produced in real business scenarios. Common errors made by human analysts included the following:

- Mapping elements to overly general classes (equivalent to *Thing*).
- Mapping elements to subtypes even when the superclass was the appropriate match. As an example, *Hierarchy* was mapped to *HierarchyType* when *Hierarchy* existed in the other model.
- Mapping elements that were simply invalid or wrong.

We encoded 6 different heuristics to flag suspicious mappings, including heuristics that can identify common errors made by our own algorithm (e.g., the tendency to match across elements with duplicate, copied documentation). The Lint engine for model mappings is thus incorporated as a key filter for semi-automated model mapping tool, to reduce the number of false positives that the human analyst needs to examine. A second use of our tool is of course to review the quality of human mappings in cases where the model mappings were produced manually.

Our key contributions are as follows:

- We describe a technique to extend existing techniques in ontology mapping to the problem of model mapping across UML, ER, and EMF models. Unlike existing approaches in schema mapping, we exploit semantic information embedded in documentation along with semantic and lexical information to perform the mapping.
- We describe a novel Lint engine which can be used to review the quality of model mappings produced either by a human or by our algorithm.
- We perform a detailed evaluation of the semi-automated tool on 7 real world model mappings. Four of the seven mappings had human mappings that were performed in a business context. We evaluated the Lint engine on these 4 mappings. The mappings involved large industry specific framework models with thousands of elements in each model in the domains of healthcare, insurance, and banking, as well as customer models in the domains of healthcare and banking. Our approach has therefore been validated on mappings that were performed for real business scenarios. In all cases, we validated the output of both tools with domain experts.

2 Related Work

Ontology matching or the related problem of schema matching is a well studied problem, with a number of different approaches that are too numerous to be outlined here in detail. We refer the reader instead to surveys of ontology or schema matching [4–6]. A sampling of ontology matching approaches include GLUE [7], PROMPT [8], HCONE-merge [9] and SAMBO [10]. Sample approaches to schema matching include Cupid [11], Artemis [12], and Clio [13–16]. Our work is mostly closely related to Falcon-AO [1, 2] and PRIOR [3], two recent approaches to ontology matching that combine some of the advantages of earlier approaches

such as linguistic and structural matching incorporated within an information-retrieval approach, and seem well positioned to be extended to address matching in shallow-structured models such as UML, ER and EMF models. Both Falcon-AO and PRIOR have been compared with existing systems in OAEI 2007 and appear to scale well in terms of performance. Because our work addresses matching across very large UML, ER and EMF data models (about 5000 elements), we adapted the approaches described in Falcon-AO and PRIOR to these models. Matching or alignment across these models is different from matching ontologies, because the semantics of these models are poorly defined compared to those of ontologies. More importantly, we report the results of applying these techniques to 7 real ontology matching problems in the field, and describe scenarios where the approach is most effective.

3 Overall Approach

3.1 Matching algorithm

Casting the matching problem to an IR problem Similar to approaches outlined in Falcon-AO [1],[2] and PRIOR ([3]), a fundamental principle in our approach is to cast the problem of model matching into a classical Information Retrieval problem. Model elements (e.g. attributes or classes) from various modeling representations (e.g. XML Schema, UML, EMF, ER) are transformed into virtual documents. A virtual document consists of one or more fields capturing the structural, lexical and semantic information associated with the corresponding model element.

A Vector Space Model (VSM) [17] is then adopted: each field F of a document is represented as a vector in a N_F-dimensional space, with N_F denoting the number of distinct words in field F of all documents. Traditional TF-IDF (Term Frequency - Inverse Document Frequency) values are used as the value of coordinates associated to terms. Formally, let D_F denotes the vector associated with the field F of a virtual document D, and $D_F[i]$ denotes the ith coordinate of the vector associated with the field F of a virtual document D:

$$D_F[i] = tf_i * idf_i \tag{1}$$

$$tf_i = |t_i|/N_F \tag{2}$$

$$idf_i = 1 + log(ND/d_i) \tag{3}$$

where

- $|t_i|$ represents the number of occurrence, in the field F of document D, of the term t corresponding to the ith coordinate of the vector D_F,
- ND corresponds to the total number of documents, and
- d_i is the number of documents in which t appears at least once in F

The similarity $sim(A, B)$ between two model elements A and B is computed as the weighted mean of the cosine of the angle formed by their field vectors.

Formally, let D and D' be the virtual documents corresponding to A and B, respectively. Let q be the number of distinct field names in all documents.

$$sim(A, B) = \frac{\sum_{k=1}^{q} \alpha_k * cosine(D_{F_k}, D'_{F_k})}{\sum_{k=1}^{q} \alpha_k} \tag{4}$$

$$cosine(D_{F_k}, D'_{F_k}) = \frac{\sum_{i=1}^{N_{F_k}} D_{F_k}[i] * D'_{F_k}[i]}{|D_{F_k}| * |D'_{F_k}|} \tag{5}$$

$$|D_F| = \sqrt{\sum_{i=1}^{N_F} (D_F[i])^2} \tag{6}$$

where α_k is the weight associated with the field F_k, which indicates the relative importance of information encoded by that field.

In our Lucene[3]-based implementation, before building document vectors, standard transformations, such as stemming/lemmatization, stop words removal, lowercasing, etc, are performed. In addition to these standard transformations, we also convert camel case words (e.g. "firstName") into corresponding group of space separated words (e.g. "first name").

Transforming model elements into virtual documents A key step in our approach is the transformation of elements of a data model into virtual documents. For simplicity of the presentation, we assume that the data model is encoded as a UML Class diagram[4]

The input of the transformation is a model element (e.g. attribute, reference/association, or class). The output is a virtual document with the the following fields:

- *name.* This field consists of the name of the input element.
- *documentation.* This field contains the documentation of the input model element.
- *containerClass.* For attribute, reference and association, this field contains the name and documentation of their containing class.
- *path.* This field contains the path from the model root package to the model element (e.g. for an attribute "bar" of the class "foo" located in the package "example", the path is /example/foo/bar).
- *body.* This field is made of the union of terms in all fields except path.

While the first two fields encode only lexical information, the next two fields (containerClass and path) capture some of the structure of the modeling elements. In our implementation, when the models to be compared appear very similar, which translates to a very large number of discovered mappings, we typically empirically adjust upwards the weight of the "containerClass" and "path" fields to convey more importance to the structural similarity.

[3] http://lucene.apache.org/java/docs/

[4] Our implementation is able to handle more data model representations, including XML Schemas, ER diagrams, and EMF ECore models.

For the simple UML model shown in Figure 3.1, 5 virtual documents will be created, among which is the following:

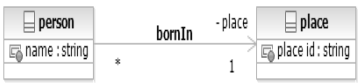


Fig. 1. Simple Model Example

1. Virtual document corresponding to the class "Place":
   - *name* : "Place"
   - *documentation*: "a bounded area defined by nature by an external authority such as a government or for an internal business purpose used to identify a location in space that is not a structured address for example country city continent postal area or risk area a place may also be used to define a logical place in a computer or telephone network e.g. laboratory e.g. hospital e.g. home e.g. doctor's office e.g. clinic"
   - *containerClass*: ""
   - *path*: "/simple test model/place"
   - *body*:"place, a bounded area defined by nature by an external authority such as a government or for an internal business purpose used to identify a location in space that is not a structured address for example country city continent postal area or risk area a place may also be used to define a logical place in a computer or telephone network e.g. laboratory e.g. hospital e.g. home e.g. doctor's office e.g. clinic"
2. Virtual document corresponding to the attribute "Place id":
   - *name* : "place id"
   - *documentation*: "the unique identifier of a place"
   - *containerClass*: "place, a bounded area defined by nature by an external authority such as a government or for an internal business purpose used to identify a location in space that is not a structured address for example country city continent postal area or risk area a place may also be used to define a logical place in a computer or telephone network e.g. laboratory e.g. hospital e.g. home e.g. doctor's office e.g. clinic"
   - *path*: "/simple test model/place/place id"
   - *body*: "place id, the unique identifier of a place, place, a bounded area defined by nature by an external authority such as a government or for an internal business purpose used to identify a location in space that is not a structured address for example country city continent postal area or risk area a place may also be used to define a logical place in a computer or telephone network e.g. laboratory e.g. hospital e.g. home e.g. doctor's office e.g. clinic"

Adding lexical and semantic similarity between terms The cosine scoring scheme presented above (4) is intolerant to even minor lexical or semantic variations in terms. For example, the cosine score computed using equation (4) for the document vectors (gender: 1, sex: 0) and (gender:0, sex: 1) will be 0 although "gender" mentioned in the first document is clearly semantically related to "sex" appearing in the second document. To address this limitation, we modify the initial vector to add, for a given term t, the indirect contributions of terms related to t as measured by a term similarity metric. Formally, instead of using D_{F_k} (resp. D'_{F_k}) in equation (4), we used the document vector $\widehat{D_{F_k}}$ whose coordinates $\widehat{D_{F_k}}[i]$, for $1 \leq i \leq N_{F_k}$, are defined as follows:

$$\widehat{D_{F_k}}[i] = D_{F_k}[i] + \beta_i * \sum_{j=1 \ \& \ j \neq i}^{N_{F_k}} termSim(t_i, t_j) * D_{F_k}[j] \tag{7}$$

$$\beta_i = \begin{cases} 0 & \text{if, for all } j \neq i, D_{F_k}[j] = 0, \\ \frac{1}{\sum_{j=1 \ \& \ j \neq i \ \& \ D_{F_k}[j] \neq 0}^{N_{F_k}} 1} & \text{otherwise} \end{cases} \tag{8}$$

where

- $termSim$ is a term similarity measure such as Jaccard or Levenshtein similarity measure (for lexical similarity), a semantic similarity measure based on WordNet [18] [19], or a combination of similarity measures. $termSim(t_i, t_j) * D_{F_k}[j]$ in (7) measures the contribution to the term t_i of the potentially related term t_j.
- β_i is the weight assigned to indirect contributions of related terms.

For efficiency, when comparing two document vectors, we only add in the modified document vectors, the contributions of terms corresponding to at least one non-zero coordinate of any of the two vectors.

The equation (7) applied to the previous example transforms (gender:1, sex :0) to (gender: 1, sex: termSim("sex", "gender")) and (gender: 0, sex: 1) to (gender: termSim("gender", "sex"), sex: 1). Assuming that termSim("sex", "gender"), which is the same as termSim("gender", "sex"), is not equal to zero, the cosine score of the transformed vectors will obviously be different from zero, and will reflect the similarity between the terms "gender" and "sex".

For the results reported in the evaluation section, only the Levenshtein similarity measure was used. Using a semantic similarity measures based on wordnet significantly increasing the algorithm running time with a marginal improvement of quality of the resulting mappings. The running time performance of semantic similarity measures based on WordNet, was still unacceptable after restricting related terms to synonyms and hyponyms.

Our approach provides a tigher integration of cosine scoring scheme and a term similarity measure. In previous work, e.g. Falcon-AO[2], the application of the term similarity measure (Levenshtein measure in Falcon-AO) is limited to names of model elements, and the final score is simply a linear combination of the cosine score and the measure of similarity between model element names.

4 Evaluation of Model Matching Algorithm

To evaluate the model matching algorithm, we accumulated industry models and customer data models from IBM architects who regularly build solutions for customers. The specific model comparisons we chose were ones that IBM architects need mapped in the field. In four cases out of 7 model matching comparisons, the matching had been performed by IBM solutions teams manually. We tried to use these as a 'gold standard' to evaluate the model matching algorithm, but unfortunately found that in 3 of 4 cases, the quality of the manual model matching was exceedingly poor. We address this issue with a tool to assess matching quality in the next section.

As shown in Table 1, the industry models we used in the comparisons included BDW (a logical data model for financial services), HPDM (a logical data model for healthcare), MDM (a model for the IBM's solution for master data management), RDWM (a model for warehouse solutions for retail organizations), and IAA (a model for insurance). Model A in the table is a customer ER model in the healthcare solutions space, model B is a customer logical data model in financial services, and model C is customer logical data model in retail. To evaluate our model matching results, we had two IBM architects assess the precision of the best possible match produced by our algorithm. Manual evaluation of the matches was performed on sample sizes of 100 in 5 of 7 cases (all cases except the IAA-BDW and A-HPDM comparisons). For IAA-BDW, we used a sample size of 50 because the algorithm produced less than 100 matches. For A-HPDM, we relied on previously created manual mappings to evaluate both precision and recall (recall was at 25%). The sizes of these models varied from 300 elements to 5000 elements.

We make two observations about our results:

- (a) The results show a great deal of variability ranging from cases where we had 100% precision in the top 100 matches, to 52% precision. This reflected the degree to which the models shared a common lineage or common vocabulary in their development. For example, RDWM was actually derived from BDW, and this is clearly reflected in the model matching results. IAA and BDW target different industries (and therefore do not have much in common), and this is a scenario where the algorithm tends to make more errors. We should point out that although IAA and BDW target different industries (insurance and banking respectively), there is a real business need for mapping common or overlapping concepts across these disparate models, so the matching exercise is not a purely academic one.
- (b) Even in cases where the precision (or recall) was low, the IBM architects attested to the utility of such a semi-automated approach to model matching, because their current process is entirely manual, tedious and error prone. None of the model mapping tools available to them currently provide results that are usable or verifiable.

Models Compared	Number of matches	Precision
A-HPDM	43	67%
B-BDW	197	74%
MDM-BDW	149	71%
MDM-HPDM	324	54%
RDWM-BDW	3632	100%
C-BDW	3263	96%
IAA-BDW	69	52%

Table 1. Model matching results

4.1 Lint Engine

We turn now to another aspect of our work, which is to somehow measure the quality of ontology matching in the field. As mentioned earlier, we initially started our work with the hope of using manual matchings as a gold standard to measure the output of our matching algorithm, but were surprised to find a rather large number of errors in the manually generated model mappings. A lot of these errors were presumably due to the ad hoc nature of the manual mapping process, leading to poor transcription of names, e.g., changes in spaces, appending package names etc. when writing mapping results in a separate spreadsheet; specification of new classes/attributes/relationships to make up a mapping, when the elements did not exist in the original models etc. Also, there were cases in which mappings were made to an absurdly generic class (such as *Thing*) which rendered them meaningless.

In order to deal with the above issues, and also improve the accuracy of our mapping tool, we decided to write a Lint Engine to detect suspicious mappings. The engine runs through a set of suspicious mapping patterns, with each pattern being assigned a severity rating and a user-friendly explanation, both specified by the domain expert. We have currently implemented the following six mapping patterns based on discussions with a domain-expert:

- *Element not found*: The pattern detects mappings where one or more elements involved does not exist in any of the models. This pattern is assigned a high severity since it indicates something clearly suspicious or wrong.
- *Exact name mismatches*: Detects mappings where a model element with an exact lexical match was not returned. This does not necessarily indicate an incorrect mapping, however does alert the user of a potentially interesting alternative that may have been missed.
- *Duplicate documentation*: Detects mappings where the exact same documentation is provided for both elements involved in the mapping. This may arise when models or portions of models are copy/pasted across.
- *Many-to-1 or 1-to-Many*: Detects cases where a single element in one model is mapped to a suspiciously large number elements in another model. As mentioned earlier, these typically denote mappings to an absurdly generic class/relation.

– *Class-Attribute proliferations*: Detects cases when a single class' attributes/relations are mapped to attributes/relations of several different classes in the other model. What makes this case suspicious is that model mappings are a means to an end, typically used to specify instance transformations. Transformations can become extremely complex when class-attribute proliferations exist.
– *Mapping without documentation*: Detects cases where all the elements involved in the mapping have no associated documentation. This could arise due to lexical and structural information playing a role in the mapping, however the lack of documentation points to a potentially weaker match.

We applied our Lint engine to the manual mappings to see if it could reveal in more detail the defects we had observed. The results are summarized in the Tables 2 - 5 below.

Total number of mappings	306
Total number of suspicious mappings	151 (51 %)
One To Many Mappings	143 (46 %)
Mapping Without Documentation	40 (25 %)
Exact Name Not Match	13 (8 %)
Duplicate Documentation	2 (1 %)

Table 2. Evaluation of B-BDW manual mappings using our Lint Engine

Total number of mappings	702
Total number of suspicious mappings	702 (100 %)
Name Not Found in Models	702 (100 %)
Mapping Without Documentation	702 (100 %)
Exact Name Not Match	30 (4 %)
One To Many Mappings	312 (44 %)

Table 3. Evaluation of BDW-MDM manual mappings using our Lint Engine

Total number of mappings	117
Total number of suspicious mappings	95 (81 %)
Mapping Without Documentation	95 (100 %)
One To Many Mappings	10 (10 %)
Duplicate Documentation Checker	9 (9 %)
Name Not Found in Models	2 (2 %)

Table 4. Evaluation of A-HPDM manual mappings using our Lint Engine

Total number of mappings	748
Total number of suspicious mappings	748 (100 %)
Mapping Without Documentation	741 (99 %)
Name Not Found in Models	459 (61 %)
Class Attribute Mapping Proliferation	472 (63 %)
Duplicate Documentation Checker	378 (50 %)
One To Many Mappings	321 (42 %)
Exact Name Not Match	33 (4 %)

Table 5. Evaluation of MDM-HPDM manual mappings using our Lint Engine

The results are quite shocking, e.g., in the BDW-MDM case, all 702 mappings specified an element that did not exist in either of the two models. The only explanation for this bizarre result is that mapping exercises, typically performed in Excel etc, are hideously inaccurate - in particular, significant approximation of the source and target elements is pervasive. Another point to note is that humans like to try and cheat and map at a generic level, and this practice seems to be quite pervasive, as such mappings were discovered in almost all the cases. Finally, the lack of, or duplication of documentation can be identified in many ways (e.g. products such as SoDA from Rational[5]) - but surfacing this during the mapping validation is very helpful. It helps present an estimation of the degree of confidence in the foundation of the mapping - the understanding of the elements being mapped.

The results were analyzed in detail by a domain expert who verified that the accuracy and usefulness for the suspicious mappings was very high (in the B-BDW case, only 1 suspicious mapping produced by Lint was actually correct). The fact that the lint engine found roughly less than 1 valid mapping for every 10 suspicious ones is an indication of the inefficiency of manual mapping practices. What the engine managed to do effectively is to filter from a huge pool of mappings, the small subset that need human attention, while hinting to the user what may be wrong by nicely grouping the suspicious mappings under different categories.

References

1. Jian, N., Hu, W., Cheng, G., Qu, Y.: Falcon-ao: Aligning ontologies with falcon. In: Proceedings of K-CAP Workshop on Integrating Ontologies. (2005)
2. Qu, Y., Hu, W., Cheng, G.: Constructing virtual documents for ontology matching. In: Proceedings of the 15th international conference on World Wide Web, Edinburgh, UK (2006)
3. Mao, M., Peng, Y., Spring, M.: A profile propagation and information retrieval based ontology mapping approach. In: Proceedings of the 3rd International Conference on Semantics, Knowledge and Grid (research track), Xian, China (2007)
4. Noy, N.F.: Semantic integration: a survey of ontology-based approaches. SIGMOD Rec. **33**(4) (2004) 65–70

[5] http://www-01.ibm.com/software/awdtools/soda/index.html

5. Kolaitis, P.G.: Schema mappings, data exchange, and metadata management. In: Proceedings of the 24th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Baltimore, Maryland (2005)
6. Bernstein, P.A., Melnik, S.: Model management 2.0: manipulating richer mappings. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, Beijing, China (2007)
7. Doan, A., Madhavan, J., Dhamankar, R., Domingos, P., Halevy, A.: Learning to match ontologies on the semantic web. The VLDB Journal **12**(4) (2003) 303–319
8. Noy, N.F., Musen, M.A.: Prompt: Algorithm and tool for automated ontology merging and alignment. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, Austin, Texas, USA (2000)
9. Kotis, K., Vouros, G., Stergiou, K.: Capturing semanticstowards automatic coordination of domain ontologies. In: AIMSA. (2004) 22–32
10. Lambrix, P., Tan, H.: Sambo-a system for aligning and merging biomedical ontologies. Web Semant. **4**(3) (2006) 196–206
11. Madhavan, J., Bernstein, P.A., Rahm, E.: Generic schema matching with cupid. In: VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2001) 49–58
12. Castano, S., De Antonellis, V., De Capitani di Vimercati, S.: Global viewing of heterogeneous data sources. IEEE Trans. on Knowl. and Data Eng. **13**(2) (2001) 277–297
13. Miller, R.J., Haas, L.M., Hernández, M.A.: Schema mapping as query discovery. In: Proceedings of 26th International Conference on Very Large Data Bases, Cairo, Egypt (2000)
14. Miller, R.J., Hernández, M.A., Haas, L.M., Yan, L.L., Ho, C.T.H., Fagin, R., Popa, L.: The clio project: Managing heterogeneity. SIGMOD Record **30**(1) (2001)
15. Bernstein, P.A., Ho, H.: Model management and schema mappings: Theory and practice. In: Proceedings of the 33rd International Conference on Very Large Data Bases, University of Vienna, Austria (2007)
16. Hernández, M.A., Popa, L., Ho, H., Naumann, F.: Clio: A schema mapping tool for information integration. In: Proceedings of the 8th International Symposium on Parallel Architectures, Algorithms and Networks, Las Vegas, Nevada, USA (2005)
17. Raghavan, V.V., Wong, S.K.M.: A critical analysis of vector space model for information retrieval. Journal of the American Society for Information Science **37**(5) (January 1999) 279–287
18. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. CoRR **cmp-lg/9709008** (1997)
19. Lin, D.: An information-theoretic definition of similarity. In: ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1998) 296–304

Mapping-Chains for studying Concept Shift in Political Ontologies

Shenghui Wang[1,2], Stefan Schlobach[2], Janet Takens[1], and Wouter van Atteveldt[1]

[1] Department of Communication Science
[2] Department of Computer Science
Vrije Universiteit Amsterdam

Abstract. For some years now ontologies have been used in Social Science, *e.g.* , in annotation of newspaper articles for disambiguating concepts within Media Analysis. These ontologies and annotations have now become objects of study in their own right, as they implicitly represent the shift of meaning of political concept over time. Manual mappings, which are intrinsically intensional, can hardly capture such subtle changes, but we claim that automatic instance-based mappings, with their extensional character, are more suitable for producing interesting **mapping-chains.**
In this paper, we evaluate the use of instance-based ontology mappings for producing concept chains in a case-study in Communication Science on a corpus with ontologies describing the Dutch election campaigns since 1994. This initial research shows the potential of the associative character of extensional mapping-chains, but also indicates a number of unsolved open questions, most significantly the lack of a proper methodology for evaluating such chains due to the open, explorative character of the task.

1 Introduction

Since 1994 Communications Scientists at the Vrije Universiteit Amsterdam have been annotating newspaper articles with controlled vocabularies (of increasing expressiveness) quantitatively studying the influence of the Media on the political processes. The idea is to code the *meaning* of sentences and articles in a formalised graph representation (called NET) similar to RDF triples. In these triples actors and issues are taken from an ontology, and the predicate usually represents opinions and moods. During recent election campaigns all newspaper articles on Dutch politics were manually coded using the NET method, with a different ontology used in each of the elections. Each ontology is more or less an adaptation of a previous one, with different foci, as in each election new issues emerged and the (societies' and scientists') view on issues changed.

As now several of these campaign data sets can easily be queried, also through the use of Semantic Web technology, Communication Scientists' interests started to also include temporal shifts of political development. In an initial analysis political developments over time were studied by querying the NET representation

of the articles from the different campaigns, which required manual mappings between the ontologies. In this paper, we propose a different approach, namely to study *concept shift* by using chains of extensional, *i.e.* , instance-based, mappings. Our hypothesis is that these mapping-chains represent subtle changes in meaning of the related concepts over time, and in this paper we will investigate this claim.

Methodology Following our previous work [1] we use information retrieval techniques to calculate document similarity between annotated articles, which we use subsequently to identify similarity between concepts. Chains between the most similar of these concepts thus produce graph-like structures (lines, trees, or DAGs), which "tell their own" story of Dutch politics over the past 15 years.

Research questions There are two major research questions, one regarding the correctness of our hypothesis, the second concerning the validity of our approach. More concretely we have to investigate:

- **RQ1:** What are suitable structures for representing mapping chains?
- **RQ2:** Can instance-based ontology mapping provide *useful* mapping-chains expressing concept shift, and how can we evaluate those chains?

The second research questions relates to a fundamental methodological issue, for which no simple answers exist: the vague character of the success criteria *usefulness*. To answer **RQ2** we will argue for usefulness through qualitative evidence, namely by providing some detailed analyses of chains in a real use-case. Automatically evaluating quality of chains is even more difficult. Remember that we want to use the extensional semantics of the concepts for determining the mappings, which makes the only comparison we have, namely an intensional gold-standard, difficult to justify. In our use-case, the line between what was identified to be a correct extensional mapping and what was an incorrect association was very fine, and studying this friction will be in our view an important future topic for this type of research.

Data, experiments and evaluation For our experiments we used 5 different ontologies from the Dutch election campaigns in 1994, 1998, 2002, 2003 and 2006. Our experiment were conducted by mapping each of ontologies with each other. Each ontology was used to annotate (around 5000) newspaper articles of the respective campaign. Some initial formal evaluation was done by comparing mappings with an existing manually created (intensional) alignment. Evaluating the quality of the chains is more tricky, as we will discuss in Section 5.3. The answer to **RQ2** therefore remains anecdotal, and finally rather unsatisfactory.

Applications and generality Capturing meaning shift, particularly the extensional associations, of concepts over time, can help communication scientists to apply analysis on the dynamics of the political developments over time. This line of research is also generalisable in many other areas where similar problems occur, such as development of medical systems, e-Science, knowledge management, and other social networks, *etc.*

2 Instance-based matching method

Instance-based ontology matching techniques have shown its capacity of dealing with matching cases where lexical and structural techniques could not be applied effectively [2, 3]. A straightforward method is to measure the common extension of concepts [4, 5]. The major limitation of this method is usually a lack of shared instances. Recently, we have investigated ways of detecting concept correlation using the similarity between their instances [2, 1]. In our case, coders used concepts to describe the content of newspaper articles. We consider an article as an instance of a concept, if the concept is used to describe this article. Our hypothesis is that, even if the ontologies during different election periods are different, two similar articles should have been coded using similar concepts. Therefore, finding similar instances can lead to similar concepts.

Let O_1 and O_2 be two ontologies which are used to annotate two instance sets I_1 and I_2. The instance-matching based method consists of two steps:

- Instance enrichment. For each instance i_1 in I_1, find the most similar instance j_2 in I_2. We consider i_1 to be an instance of the concepts which j_2 is described with. The same operation is applied in the other direction. In the end, an artificial common instance set is built.
- Concept matching. Each concept corresponds to a set of instances, including their real instances and those enriched in the previous step. A corrected Jaccard similarity measure is applied to calculate the similarity between concepts from different years. That is

$$\text{Jacc} = \frac{\sqrt{|{c_1}^i \cup {c_2}^i| * (|{c_1}^i \cup {c_2}^i| - 0.8)}}{|{c_1}^i \cup {c_2}^i|} \tag{1}$$

  where ${c_1}^i$, ${c_2}^i$ are the instance sets of two concept $c_1(\in O_1)$ and $c_2(\in O_2)$. [3]

Two concepts with sufficient similarity are considered mapped. A set of mappings between concepts of two ontologies form an alignment between the ontologies.

Instance matching There are different ways to match instances. A simple method is to consider instances as documents, and apply information retrieval techniques to retrieve similar instances (documents). We use a tf-idf weighting scheme which is often exploited in the vector space model for information retrieval and text mining [6]. The idea is that each document is represented by a vector, each element is a weight of a word which occurs in this document. Each word is weighted using its $tf\text{-}idf$ value. Traditionally, a query is represented as a vector using the idf of the to-be-queried dataset. In our case, the same word is likely to have different importance in different datasets, therefore, while building the vector representation of each document, we use the corresponding idf values

[3] To avoid very high scores in the case of very few instances a 0.8 parameter was chosen so that concepts with a single (also shared) instance obtain the same score as concepts with, in the limit, infinitely many instances, 20% of which co-occur.

Year	Articles	Concepts used	Concept manually mapped to	
			existing concepts	added news concepts
1994	1502	101	54	37
1998	5635	312	154	135
2002	6323	370	201	110
2003	5053	299	190	89
2006	5126	580		

Table 1. Datasets and manual mappings of different years.

of words calculated within the dataset to which the document belongs. Based on such vectors, the cosine similarity is used to determine the similarity between two documents. In this way, instances from different datasets are matched, and the information is used for the enrichment process.

Chains of mappings After alignments are generated between multiple ontologies, with some ontologies involved in multiple alignments, it is possible to generate chains of mappings between a series of ontologies, in, for example, a chronological order.

Let A_{12} and A_{23} be the alignments between O_1 and O_2 and between O_2 and O_3. If there is a mapping in A_{12}, $< c_{1i}, c_{2j}, v_{ij}^{12} >$ and a mapping in A_{23}, $< c_{2j}, c_{3k}, v_{jk}^{23} >$, this results in a two-step chain of mapping from c_{1i} to c_{3k} via c_{2j}, with a confidence value $v_{ik} = v_{ij}^{12} \times v_{jk}^{23}$. When there are a series of alignments between O_1 and O_2, O_2 and O_3, until O_{n-1} and O_n, this will result in n-1-step chains of mappings

$$< c_{1i} \rightarrow c_{2j} \rightarrow \cdots \rightarrow c_{n-1,k} \rightarrow c_{nl}, v_{ij}^{12} \times \ldots v_{kl}^{n-1,n} > .$$

In this paper, we investigate 4 different kinds of mapping-chains, and investigate their usefulness in a practical application:

1. Top-1 forward chain (such as in Fig.:2): in each step, only the mapping with highest confidence is considered.
2. Top-n forward-chains (such as in Fig.:4): in each step, the the best n mappings are considered, starting with the first ontology.
3. Top-n backward-chains (such as in Fig.:5): in each step, the the best n are considered, starting with the last ontology.
4. Top-n kite (such as in Fig.:6): starting with the first ontology in each step, the mappings with the n highest confidence values for which there exist a top n mapping chain to the correct mapping (according to the gold standard).

3 Ontologies in Semantic Network Analysis

The series of ontologies to be mapped are the ontologies used to code newspapers during five recent Dutch elections taking place in 1994, 1998, 2002, 2003, and 2006. The articles were coded using the Network analysis of Evaluative Texts (NET) method [7], popularly used in the Semantic Network Analysis. During

each election year, annotators coded newspaper articles using the ontology available to them. Take as an example a sentence in a newspaper article during the election period in 2006.

Example 1. Het Openbaar Ministerie (OM) wil de komende vier jaar mensenhandel uitroeien. (*The Justice Department (OM) wants to eliminate human trafficking within the next four years.*)

The sentence is coded as `<om, -1,human trafficking>`, where `om` and `human trafficking` are two concepts in the ontology used in 2006, while `-1` indicates the Justice Department is negative about human trafficking. In this example, we consider this sentence to be an instance of the two concepts involved. All five ontologies are represented in the standard SKOS format [8]. Each concept has an prefLabel and possibly a few altLabel which are the synonyms of this concept and also used by coders in the coding process. Except the most recent election, all the newspapers are coded at the article level, but mainly based on the first three sentences. In 2006, the coding is at the sentence level.

The synonymous concepts were found manually. As shown in Table 1, the number of manually mapped concepts is smaller than that of concepts found in the actual coding. The reason is that new variations of the concepts were manually input to the database. These variations are very likely to be synonyms of concepts in the ontologies or pure typos, which were not covered during the manual mapping process.

Alignments between ontologies of previous years to the latest 2006 version have also been made manually. However, some concepts used in previous years cannot find the exact correspondences in the 2006 version. In that case, the domain experts added new concepts to the current version. The last column of Table 1 indicates how many new concepts were added during the manual aligning process. These new concepts are not used during the coding of 2006 articles, which means they do not have any instances in the 2006 corpus and were therefore not considered in our automated evaluation.

4 Base experiments: single mappings

Before focussing on chains of mappings, we need to show that our methods for calculating individual mappings are trustworthy. We first map all previous ontologies to the 2006 ontology. According to the extensional mapping technique, one concept can be related to multiple concepts, each with a certain amount of relatedness. As only one mapping for each concept was considered in the manual mapping results, for each concept, we take the mapping with the highest confidence value (*i.e.* , the corrected Jaccard similarity) as the final mapping of this concept. Table 2 shows precision and recall of these 1:1 mappings.

For all the concepts which were manually mapped with existing concepts in the 2006 ontology, we also measure the mean reciprocal rank (MRR) $mrr = \frac{1}{|C|}\sum_{i=1}^{C}\frac{1}{rank_i}$, where C is the set of concepts, the $rank_i$ is the rank of the concept which C_i should be mapped to. When C_i does not have a match, the

Year	Precision	Recall
1994	0.22	0.22
2002	0.36	0.35

Year	Precision	Recall
1998	0.33	0.33
2003	0.4	0.4

Table 2. Evaluation of 1:1 mappings based on the sentence and article level

Year	Concepts Found	Concepts matched	MRR
1994	532	39	0.41
2002	570	143	0.54

Year	Concepts Found	Concepts matched	MRR
1998	561	102	0.47
2003	570	136	0.58

Table 3. Mean Reciprocal Rank: from 2006 to the previous years, where "Concepts Found" is the number of concepts for which we have found some mappings, "Concepts matched" is the number of concepts for which we have recovered the correct mapping.

reciprocal rank is set to 0. A higher *mrr* indicates the correct matches are ranked in the more front position. Table 3 shows that the correct mapping is ranked on average within the top 10 proposed ones.

5 Main experiments: Chains of mappings

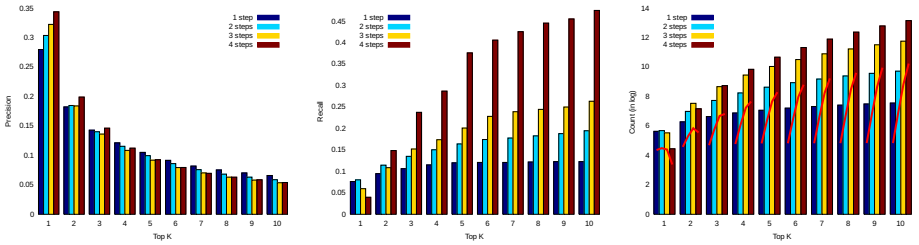
The main topic of this paper is to investigate the use of chains of mappings of concepts from ontologies from 1994 to 2006.

5.1 Quantitative analysis

Based on the manual alignments, we can measure the precision and recall of the chains. For each concept from 1994, top K mappings are taken into consideration, each of which will be expanded by its top K mappings too, and so on.[4] A chain can start from any concept of any year. For all chains with n steps, $n = 1, 2, 3, 4$, we measure the precision and recall respectively. A chain is considered to be correct if the two end-point concepts form a correct mapping according to gold standard, *i.e.* , the correctness is 1; a chain is partially correct and the correctness is the number of correct mappings on the way over the number of steps. In the end we take the average of the correctness of all individual chains as the final precision. The partial correctness is not considered when calculating the recall. The evaluation results are shown in Fig. 1.

Clearly, when considering more mapping candidates, more noisy data is included. Fig. 1 (c) gives the raw count of chains in terms of the choice of K. Note the numbers are in log scale. The red line indicates the number of chains with the two end-point concepts form a correct mapping judged by the gold standard.

[4] By expanding via different intermediate mappings from one concept, the total amount of chains is growing, but not exponentially. The reason is that concepts related to the starting concept tend to have a similar extensional semantics.



(a) Precision (b) Recall (c) Raw count

Fig. 1. Performance of the mapping chains. The x-axis indicates, for each concept, the top K related concepts are considered.

If only taking top 1, 2 or 3 related concepts, when the steps goes up, the number of correct chains drop, but the total amount of chains drops even faster. The precision of multi-step chains is actually higher than that of shorter chains. This suggests, even the absolute correctness of direct mappings may not be perfect, a multi-step chains of less perfect mappings may still lead to correct mappings over all time. Unfortunately, when taking more less related concepts, the number of correct chains goes up, but the sheer amount of total chains climbs up more rapidly, which cause the precision to drop in the end; the precision of multi-step chains is lower than that of shorter chains, that is the quality of the mappings degraded when the intermediate steps became longer.

5.2 Qualitative analysis of chains

A qualitative analysis of chains of mappings provides interesting insights in the value of the matching of ontologies over time for social sciences. A typical example of mapping of concepts over time will be discussed. The **domain-specific** political analysis will be supplemented by a methodological discussion (we call it **Metaanalysis** provided in italics) from the perspective of the usefulness of mapping-chains.

Let us start with an analysis of different chains for the two concepts asylum seekers ("asielzoekers") and senior citizens ("ouderen"). Figure 2 shows that the concept of asylum seekers (label: asielzoekers) is correctly mapped in each of the election years to the same concept at the highest rank. This result indicates that no or limited topic drift occurred. The confidence value slowly deteriorates, which might imply that the debate about asylum seekers has become more multi-faceted. An alternative explanation is that the number of concepts relating to asylum policy in the ontologies has increased because of the increasing political interest in asylum policy.

Fig. 2. Top 1 foward expansion for concept asielzoekers

Metaanalysis 1: *Even from a simple top 1 forward chain, some lessons can be drawn. However, note that analyses of this kind depend on the* ***confidence values****, which are rather* ***dubious*** *at best. Top 1 chains are more interesting objects of study when more drastic shifts occur.*

Fig. 3. Top 1 foward expansion for concept ouderen

Figure 3 shows that the elderly concept (ouderen) is only mapped to the expected concept between 1994 and 1998. The 1998 concept of the elderly is mapped to "obligation to apply for a job for the unemployed" (sollicitatieplicht). The abolition of the exemption of the obligation to apply for a job for the elderly was an election issues in 2002, which explains the link between these concepts. Both concepts should be considered as social security issues from a theoretical stance, since the elderly became an election issue during election campaigns with regard to special social security arrangement for the elderly. In 2003 the obligation to apply is correctly mapped to the 2002 concept. In 2006 the obligation to apply is mapped to the related concept of the unemployed.

Metaanalysis 2: ***Association versus similarity:*** *One of the crucial methodological problems is the formal meaning of the mappings between two concepts such as the elderly and the obligation to apply. Clearly, our instance-based methods find mappings with an extensional semantics,* i.e. *, the use of the concepts in annotating articles is related. A domain expert can identify elements of the intensional meaning that relate these two concepts in the specific case. Concretely, the issue "senior citizen" in 1998 and the issue "obligation to apply" in 2002 also share an intensional meaning. However, to the best of our knowledge there is no theory to formalise the relation between the extensional meaning and the intensional meaning of the mappings. In the following we will see examples where using association misses the goal of finding similarity, in particular when chains of mappings are considered.*

Although the asylum seeker concept is mapped correctly, the chains including top 2 concepts – represented in Figure 4 – give additional insights into the nature of the asylum debate over time. An analysis of the secondly ranked concepts shows with which issues asylum seekers have been associated during the election campaigns. In 1998 asylum seekers are mapped to the Cabinet of the time (kabinet kokmierlods), which follows from the fact that this Cabinet paid much attention to this issue. The second rank concepts in the following years indicate changes in the proposed policy measures. In 2002, when the anti-immigration party LPF came into power, asylum seekers are mapped to the constraint of the influx of refugees (instroom beperking). In 2003, after the LPF had left the government, it was mapped to a non-restrictive measure, assistance to illegal immigrants (opvang illegalen). In 2006 – the year in which another anti-immigration

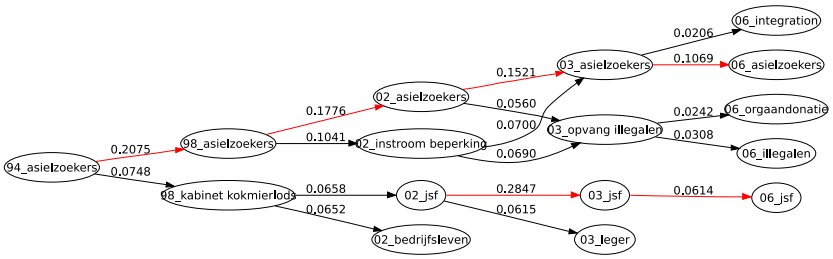


Fig. 4. Top 2 foward expansion

party made an upsurge – was mapped to integration. It is worth noting that in 2006 the second rank mapping (integration) has a positive connotation, which suggests that the new anti-immigration party did not manage to influence public opinion to the degree the LPF managed in 2002.

> **Metaanalysis 3**: *This nice example shows a number of useful findings from the Communication Scientist perspective. The* ***chains of intensional meaning of the concepts at the time of their use provides interesting, and very subtle, developments for further investigation****. Given that the top 1 mappings are all correct, the interest lies particular in the second ranked mappings of each step, which tell a story of Dutch politics.*

With the expansion of the chain, some concepts that were mapped to secondly ranked concepts do not always directly relate to asylum seekers anymore. While the secondly ranked concept assistance to illegal immigrants in 2003 was plausibly mapped to illegal immigrants (illegalen), it was also mapped to organ donation (orgaandonatie). This latter mapping is explained by a particular political actor who propagated both the assistance to illegal immigrants in 2003 and organ donation in 2006.

The lower half of the figure does not directly relate to asylum seekers either, since the Cabinet in 2003 was not mapped at a high rank to asylum seekers anymore, but to the military aircraft Joint Strike Fighter (jsf) and business (bedrijfsleven), which in turn were mapped to concepts in the military and economical area. We omitted parts of the lower half for space reasons.

> **Metaanalysis 4**: *These are examples, where association can lead to unrelated concepts in 2 steps only. This highlights one of the biggest methodological challenges: how to distinguish useful and non-useful chains.* ***Early erroneous associations can turn large parts of the analysis practically useless.***

Figure 5 containing the backward chain starting from 2006, complements the information about the nature of the asylum debate. Studying the secondly ranked concepts mapped to asylum seekers in the backward direction, partially differing association with asylum seekers appear. Asylum seekers in 2006 are mapped to crime (criminaliteit) in 2003. In recent elections immigrants (including asylum seekers) have been regularly associated with crime by anti-immigration parties. Although these concepts are not directly related, they are related to each other

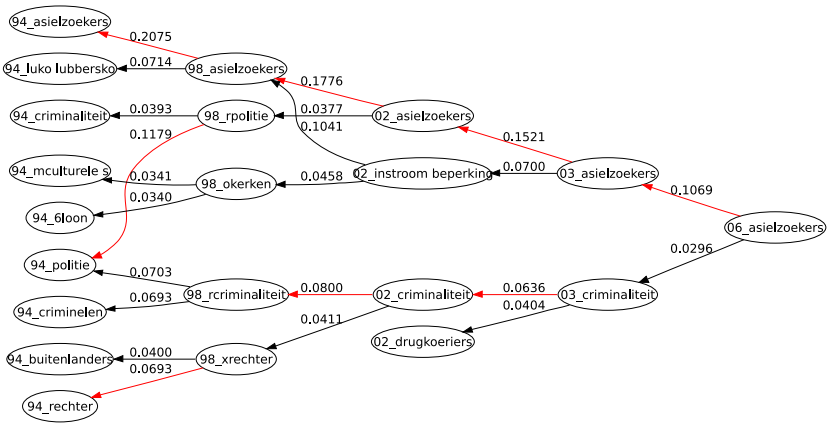


Fig. 5. Top 2 backward expansion

in the political reality. As in the forward mapping asylum seekers in 2003 is mapped to the constraint of the influx of refugees in 2002. In 1998 they are mapped to the police (rpolitie) and in 1994 to the Cabinet (luko lubbersko). The mapping to the police is in line with the mapping to crime in 2003.

> **Metaanalysis 5**: ***The chains of the concept mappings in two different directions are complementary***. *While it seems, for example, anomalous that churches (okerken) in 1998 is mapped to the constraint of the influx of asylum seekers in 2002, the mapping of the constraint of the influx of asylum seekers in 2002 to assistance to asylum seekers in 1998 in the opposite direction helps to explain this mapping, since churches played an important role in the assistance to asylum seekers.*

It is noticeable that the expansion chains do not expand exponentially, but still faster than we expected.[5] An interesting phenomenon is that mappings do not converge again, *i.e.* , once an association happens in one year, it usually does not associate back the following year to the same topic. This is an interesting finding, for which we do not have an explanation.

> **Metaanalysis 6**: ***The expansion factor is meaningful*** *in two ways: it gives an indication on the debate itself, but it can also be an indication for the mapping quality. The smaller the tree is, the more closely related are the associated concepts, which might indicate that the mapping quality is better than that for larger trees.*

The kite with two correct endpoint concepts integrates the information from the previous figures. In Figure 6, it becomes clear that the concept of asylum seekers is correctly mapped from one election year to another. Additionally it shows that the asylum debate is both associated with central concepts in the

[5] In the top 2 chains we studied the average width was 12.

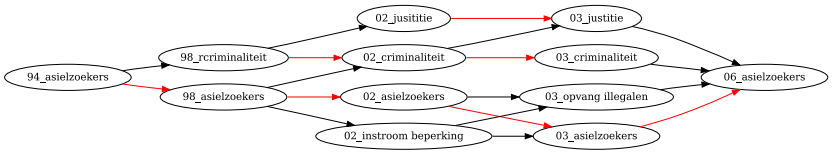


Fig. 6. Kite with two correct endpoint concepts

asylum policy debate, the constraint of the influx of refugees and the assistance to illegal immigrants, and to crime, a concept that is, however not directly related to asylum seekers, related to the concept in the political reality.

> **Metaanalysis 7**: ***Kites from end to end seem the most useful way of illustrating concept drift****, as incorrect associations are eliminated (because they are not mapped back to the start-concept). However, such a kite is less fine-grained than the forward chain: the subtle change between negative and positive connotation between 02_instroom beperking and 06_integration is lost.*

5.3 Discussion

The issues discussed in the previous section mostly center around two questions: the usefulness of our proposed mapping chains, and the meaning of the extensional mappings in the first place. Studying a representative sample of chains through our domain expert indicates clearly that mapping chains can be interesting given the reasonable quality of the individual mappings we can produce.

However, apart from the concern about the accuracy of the mapping method used here, two problems are apparent to which we do not yet have a satisfactory solution: the lack of a notion of correctness of extensional mappings in the chains, and the evaluation of this correctness. Our manual analysis shows plenty of examples where an associative semantics of mappings based on the extensions of concepts corresponds to an intensional relation of the meaning between of two concepts. However, in other examples these associations often totally diverge from what domain experts find acceptable intensional similarities. We have no intuition yet how to address this problem, *i.e.* , how to formalise and study it. The most promising solution for addressing this problem is to consider the kite structures, in which the disambiguation of mappings is achieved by requiring a mapping back to the original concept. In that way wild mismatches can be eliminated from the temporal chains. Many interesting parts of the temporal "story" get lost in this approach, though.

The second problem is the strongly related problem of evaluation: so far we found only two ways of evaluating our constructs: 1) comparing the chains with an intensional gold standard, and 2) having a domain expert evaluate each of the chains. Obviously, the first option is methodologically not valid, as we evaluate against something we know not to be the solution. The second approach is more acceptable from the domain perspective, but manual checking is very expensive

and difficult to quantify, if it is not yet impossible to identify all "intertesting chains." The only practical solutions we found so far is to use the spreading factor of the top K chains over time. In our view, the fewer leaves such a tree has, the semantically closer the mappings should be. However, this idea is build on intuition rather than empirical findings.

6 Conclusion

In this paper we introduced different representations of mapping chains and evaluated them over sequences of political ontologies in a Media study driven by Communication Scientists. We used instance-based mappings between pairs of ontologies to calculate sequences of extensional mappings and show, for a usecase analysing Dutch election campaigns, some interesting qualitative findings.

Apart from these stimulating examples we also provided an initial evaluation of our proposal, both qualitative and quantitative. However, this evaluation is tricky, as neither the notion of correctness of an extensional mapping is well-defined, nor do we have a sound evaluation methodology yet.

For us the general lessons for the ontology mapping community is twofold: that the semantics of mappings is not yet fully understood, particularly, *w.r.t.*, extensional semantics, and that mappings in a dynamic context are challenging, and worthwhile, objects of study in addition to their known static variants.

References

1. Schopman, B., Wang, S., Schlobach, S.: Deriving concept mappings through instance mappings. In: Proceedings of the 3rd Asian Semantic Web Conference, Bangkok, Thailand (2008)
2. Wang, S., Englebienne, G., Schlobach, S.: Learning concept mappings from instance similarity. In: Proceedings of the 7th International Semantic Web Conference (ISWC 2008). Volume 5318 of Lecture Notes in Computer Science., Karlsruhe, Germany, Springer (October 2008) 339–355
3. Wang, S., Isaac, A., Schopman, B., Schlobach, S., van der Meij, L.: Matching multilingual subject vocabularies. In: Proceedings of the 13th European Conference on Digital Libraries (ECDL2009), Corfu, Greece (September 2009)
4. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer Verlag (2007)
5. Isaac, A., van der Meij, L., Schlobach, S., Wang, S.: An empirical study of instance-based ontology matching. In: Proceedings of the 6th International Semantic Web Conference (ISWC 2007). Volume 4825 of Lecture Notes in Computer Science., Busan, Korea, Springer (2007)
6. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill (1983)
7. Van Cuilenburg, J., Kleinnijenhuis, J., De ridder, J.: Towards a graph theory of journalistic texts. European Journal of Communication **1** (1986) 65–96
8. Isaac, A., Summers, E.: SKOS Primer. W3C Group Note (2009)

A pattern-based ontology matching approach for detecting complex correspondences

Dominique Ritze[1], Christian Meilicke[1], Ondřej Šváb-Zamazal[2], and Heiner Stuckenschmidt[1]

[1]University of Mannheim,
dritze@mail.uni-mannheim.de, {christian, heiner}@informatik.uni-mannheim.de
[2]University of Economics, Prague, ondrej.zamazal@vse.cz

Abstract. State of the art ontology matching techniques are limited to detect simple correspondences between atomic concepts and properties. Nevertheless, for many concepts and properties atomic counterparts will not exist, while it is possible to construct equivalent complex concept and property descriptions. We define a correspondence where at least one of the linked entities is non-atomic as complex correspondence. Further, we introduce several patterns describing complex correspondences. In particular, we focus on methods for automatically detecting complex correspondences. These methods are based on a combination of basic matching techniques. We conduct experiments with different datasets and discuss the results.

1 Introduction

Ontology matching is referred to as a means for resolving the problem of semantic heterogeneity [3]. This problem is caused by the possibility to describe the same domain by the use of ontologies that differ to a large degree. Ontology engineers might, for example, chose different vocabularies to describe the same entities. There might also be ontologies where some parts are modeled in a fine grained way, while in other ontologies there are only shallow concept hierarchies in the relevant branches. These kinds of heterogeneities can be resolved by state of the art ontology matching systems, which might e.g. detect that *hasAuthor* and *writtenBy* are equivalent properties and only different vocabulary is used. Moreover a matching system might identify, that *Author* is more general as both concepts *FirstAuthor* and *CoAuthor*.

However, ontological heterogeneities are not restricted to these kind of problems: different modeling styles might require more than equivalence or subsumption correspondences between atomic concepts and properties.[1] Semantic relations between complex descriptions become necessary. This is illustrated by the following example: While in one ontology we have an atomic concept *AcceptedPaper*, in another ontology we have the general concept *Paper* and the boolean property *accepted*. An *AcceptedPaper* in the first ontology corresponds in the second ontology to a *Paper* that has been *accepted*. Such a correspondence, where at least one of the linked entities is a complex concept

[1] Atomic concepts/properties are sometimes also referred to as named concepts/properties resp. concept/property names.

or property description, is referred to as complex correspondence in the following. As main contribution of this paper we suggest an automated pattern based approach to detect certain types of complex correspondences and study its performance by applying it on different datasets. Even though different researchers were concerned with similar topics (see [11]), to our knowledge none of the resulting works was concerned with automated detection in an experimental setting. Exceptions can be found in the machine learning community (see Section 2).

We first discuss related work centered around the notion of a complex correspondence in Section 2. We then present four patterns of complex correspondences in Section 3. In Section 4 we suggest the algorithms we designed to detect occurrences of these patterns. Each of these algorithms is described as a conjunction of conditions, which are easy to check by basic matching techniques. In Section 5 we apply the algorithms on two datasets from the OAEI and show that the proposed techniques can be used to detect a significant amount of complex correspondences. We end with a conclusion in Section 6.

2 Related Work

Complex matching is a well known topic in database schema matching. In [1] the authors describe complex matches as matching corresponding attributes on which some operation was applied, e.g. a name is equivalent with concatenation of a first-name and a last-name. There are several systems dealing with this kind of database schema matching. On the other hand complex matching is relatively new in the ontology matching field. Most of the state of the art matchers just find (simple) correspondences between two atomic terms. However, pragmatic concerns call for complex matching. We also experienced this during discussions at the OM-2008. It turns out that simple correspondences are too limited to capture all meaningful relations between concepts and properties of two related ontologies. This is an important aspect with respect to application scenarios making use of alignments e.g. instance migration scenarios. There are three diverse aspects of complex correspondences: designing (defining), finding and representing them.

In [8] complex correspondences are mainly considered from design and representation aspects. Complex correspondences are captured as correspondence patterns. They are solutions for recurring mismatches being raised during aligning two ontologies. These patterns are now being included within *Ontology Design Patterns* (ODP)[2]. This work considers complex matching as task that had to be conducted by a human user, which might e.g. be a domain expert. Experts can take advantage of diverse templates for capturing complex and correct matching. However, this collection of patterns can also be exploited by some automated matching approach, as suggested and shown in this paper.

In [11] authors tried to find complex correspondences using pattern-based detection of different semantic structures in ontologies. The most refined pattern is concerned

[2] In the taxonomy of patterns at the ODP portal (`http://ontologydesignpatterns.org/wiki/OPTypes`) category AlignmentODP corresponds best with the patterns in this paper, while category CorrespondeceODP is a more general category.

with 'N-ary' relation detection. After detecting an instance of the pattern (using query language and some string-based heuristics) additional conditions (mainly string-based comparisons) over related entities wrt. matching are checked. While there are some experiments with pattern detection in one ontology, experiments with matching tasks are missing.

Furthermore, in [12] the authors consider an approach for pattern-based ontology transformation useful for diverse purposes. One particular use case is ontology matching where this method enables finding further originally missed correspondences. Ontologies are transformed according to transformation patterns and then any matcher can be applied. Authors hypothesize that matchers can work with some structures better than with others. This approach uses *Expressive alignment language*[3] based on [2] which extends the original INRIA alignment format. This language enables to express complex structures on each side of an alignment (set operators, restriction for entities and relations). Furthermore it is possible to use variables and transformation functions for transforming attribute values. "Basically, complex correspondences are employed indirectly in the ontology matching process at a pre-processing step where ontology patterns are detected and transformed [13]." Unlike, in this paper complex correspondences are detected directly taking advantage of information from not only two ontologies being aligned but also from a reference alignment composed of simple correspondences.

Regarding ontology matching, there are a few matchers trying to find complex correspondences based on machine learning approaches (see [9] for a general description). A concrete matching system is presented in [6]. These approaches take correspondences with more than two atomic terms into account, but require the ontologies to include matchable instances. However, ontologies often contain disjoint sets of instances, such that for each instance of one ontology there exists no counterpart in the other ontology and vice versa. The approach proposed in this paper does not require the existence of matchable instances at all.

3 Complex Correspondence Patterns

In the following we propose four patterns for complex correspondences that, due to a preparatory study, we expect to occur frequently within ontology matching problems. We first report about our preparatory study, followed by a detailed presentation of each pattern. Each pattern is also explained by an example depicted in Figure 1. Without explicitly mentioning it, we will refer to Figure 1 throughout this section. Further we use $\mathcal{O}_1$ and $\mathcal{O}_2$ to refer to two aligned ontologies, and we use prefix notation $i\#C$ to refer to an entity C from ontology $\mathcal{O}_i$.

First of all we had to collect different types of complex correspondences. We considered the examples found in [9] and also profited from the discussion of the consensus track at OM 2008, which highlighted the need for complex correspondences.[4] After we had a few ideas, we started observing two sets of ontologies manually to detect concrete examples for complex correspondences. The specific ontologies which we examined are the SIGKDD, CMT, EKAW, IASTED, and CONFOF ontologies of the conference dataset

[3] http://alignapi.gforge.inria.fr/language.html
[4] http://nb.vse.cz/~svabo/oaei2008/cbw08.pdf

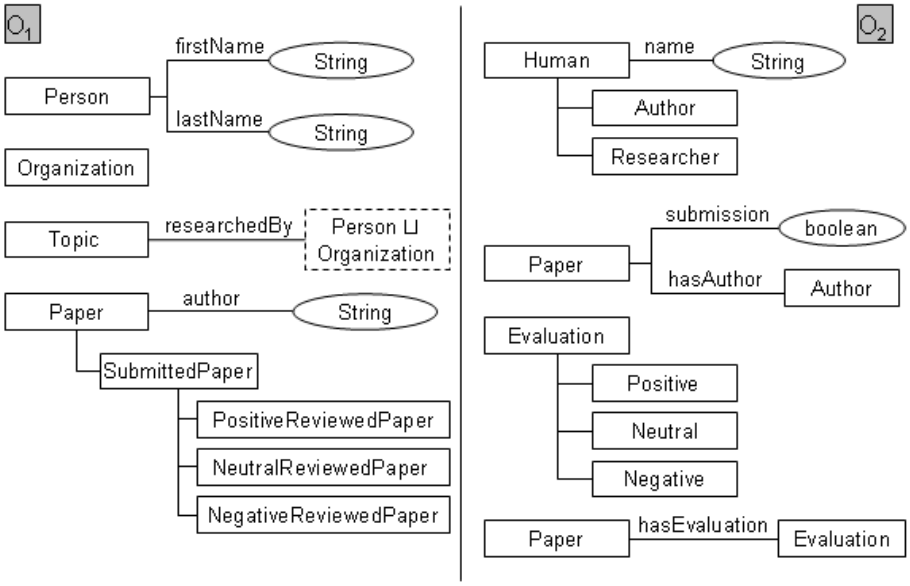


Fig. 1. Two example ontologies to explain the complex patterns

and ontologies 101, 301, 302, 303, and 304 of the benchmark track. The first dataset describes the domain of conferences. This seems to be suitable [10] because most persons dealing with ontologies are academics and know this topic already. Therefore it is easier to understand complex interdependencies in this domain instead compared to an unfamiliar domain like e.g. medical domains. The OAEI Benchmark ontologies attend the domain bibliography which is also well-known by academics. Another reason for choosing these ontologies are the existing and freely available reference alignments. For the conference dataset an alignment is available for every pair of two ontologies. Only for each combination with ontology 101 an alignment is available for the benchmark ontologies, resulting in four matching tasks. In Section 4 we will explain in how far and for which purpose a reference alignment, which consists of simple correspondences, is required.

The first three patterns are very similar, nevertheless, it will turn out that different algorithms are required to detect concrete complex correspondences. In accordance with [8] we will refer to them as *Class by Attribute Type pattern*, *Class by Inverse Attribute Type pattern*, and *Class by Attribute Value pattern*. In the following we give a formal description as well as an example for each pattern.

Class by Attribute Type pattern *(CAT)* This pattern occurs very often when we have disjoint sibling concept. In such a situation the same pattern can be used to define each of the sibling concepts.

Formal Pattern: $1\#A \equiv \exists 2\#R.2\#B$
Example: $1\#PositiveReviewedPaper \equiv \exists 2\#hasEvaluation.2\#Positive$

With respect to the ontologies depicted in Figure 1 we can construct correspondences of this type for the concepts Positive-, Neutral-, and NegativeReviewedPaper.

Class by Inverse Attribute Type pattern *(CAT^{-1})* The following pattern requires to make use of the inverse $2\#R^{-1}$ of property $2\#R$, since we want to define $1\#A$ as subconcept of $2\#R$'s range.

Formal Pattern: $1\#A \equiv 2\#B \sqcap \exists 2\#R^{-1}.\top$
Example: $2\#Researcher \equiv 1\#Person \sqcap \exists 1\#researchedBy^{-1}.\top$

Given an ontology which contains a property and its inverse property as named entities, it is possible to describe the same correspondences as *Class by Attribute Type pattern* and as *Class by Inverse Attribute Type pattern*. Nevertheless, an inverse property might often not be defined as atomic entity in the ontology or might be named in a way which makes a correct matching harder.

Class by Attribute Value pattern *(CAV)* While in the *Class by Attribute Type pattern* membership to a concept was a necessary condition, we now make use of nominals defined by concrete data values.

Formal Pattern: $1\#A \equiv \exists 2\#R.\{\ldots\}$ (where $\{\ldots\}$ is a set of concrete data values)
Example: $1\#submittedPaper \equiv \exists 2\#submission.\{true\}$

Another typical example is the distinction between *LateRegisteredParticipant* and *EarlyRegisteredParticipant*. In particular, the boolean variant of the pattern occurs to distinguish between complementary subclasses. However, in general there might be more than two relevant values. The following correspondence is a more complex example: $1\#StudentPassedExam \equiv \exists 2\#hasExamScore.\{A, B, C, D\}$.

Property Chain pattern *(PC)* [5] In the following we assume that in $\mathcal{O}_1$ property $1\#author$ relates a paper to the name of its author, while in $\mathcal{O}_2$ $2\#author$ relates a paper to its author and the datatype property $2\#name$ relates a person to its name. Under these circumstances a chain of properties in $\mathcal{O}_2$ is equivalent to an atomic property in $\mathcal{O}_1$.

Formal Pattern: $1\#R \equiv 2\#P \circ 2\#Q$
Example: $1\#author \equiv 2\#hasAuthor \circ 2\#name$

Conventional matching systems focus only on correspondences between atomic entities. Therefore, a matcher might detect a similarity between $1\#R$ and $2\#P$ and one between $1\#R$ and $2\#Q$, but will finally decide to output the one with higher similarity. This observation already indicates that state of the art matching techniques can

[5] Correspondence patterns library [8] explicitly contains *(CAT)* and *(CAV)*, other two patterns *(PC)* and *(CAT^{-1})* are not explicitly presented there.

be exploited to generate complex correspondences. In particular, we will argue in the next section, that it is possible to detect complex correspondences by combining simple techniques in an intelligent way.[6]

4 Algorithms

The techniques we are using for detecting complex correspondences are based on combinations of both linguistic and structural methods. In the following we shortly list and describe these approaches. The structural techniques require the existence of a reference alignment $\mathcal{R}$ that consists of simple equivalence correspondences between atomic concepts. In particular, it would also be possible to use a matcher generated (and partially incorrect) alignment, but in our first experiments we wanted to avoid any additional source of error.

Structural Criteria To decide whether two or more entities are related via complex correspondences, information about their position in the ontology hierarchy is required. Therefore, we have to check whether two concepts are in a subclass resp. superclass relation, or are even equivalent concepts. It might also be important to know if two concepts are non overlapping, disjoint concepts. Properties are connected to the concepts hierarchy via domain and range restrictions, which are thus also important context information. All of these notions are clearly defined within a single ontology, however, we extend these notions to a pair of aligned ontologies. $1\#C$ is also referred to as a subconcept of $2\#D$ if there exists a correspondence $1\#C' = 2\#D' \in \mathcal{R}$ such that $\mathcal{O}_1 \models 1\#C \subseteq 1\#C'$ and $\mathcal{O}_2 \models 2\#D' \subseteq 2\#D$.

Syntactical Criteria The most efficient methods used in ontology matching are based on string comparisons e.g. comparing concept id (the fragment of the concepts URI) resp. label to compute a similarity between ontological elements. We also make use of this basic method by computing a similarity measure between normalized strings based on the Levenshtein measure [4]. For the sake of simplicity we refer to the maximum value obtained from id and label comparison as label similarity in the following. For some operations we need to determine the head noun of a given compound concept/property label. Thus, we can e.g. detect that `Reviewer` is the head noun of `ExternalReviewer`. Sometimes we are simply interested in the first part of a label, sometimes in the head noun and sometimes in the remaining parts.

Data type Compatibility Two data types are compatible if one data type can be translated into the other and vice versa. This becomes relevant whenever datatype properties are involved. We determined compatibility in a wide sense. E.g. data type `String` is compatible to every other data type while `Date` is not compatible to `Boolean`.

[6] Even experts tend to avoid the introduction of complex correspondences. The property chain $1\#R \equiv 2\#P \circ 2\#Q$, for example, is sometimes reflected by one (two) correspondence(s) $1\#R \equiv 2\#P$ or (and) $1\#R \equiv 2\#Q$. See for example the reference alignment for OAEI benchmark test case 301 where $101\#date \equiv 301\#hasYear$ and $101\#year \equiv 301\#hasYear$ which should be replaced by $101\#date \circ 101\#year \equiv 301\#hasYear$.

A more detailed description can be found in [7]. Overall we emphasize that our methodology does not exceed basic functionalities which we normally would expect to be part of any state of the art matching system.

Class by Attribute Type pattern A correspondence $1\#A \equiv \exists 2\#R.2\#B$ of the *CAT* type is generated by our algorithm, if all following conditions hold.

1. The string that results from removing the head noun from the label of $1\#A$ is similar to the label of $2\#B$.
2. There exists a class $2\#C$ that is a superclass of $2\#B$, range of $2\#R$ and has also a label similar to $2\#R$.
3. The domain of $2\#R$ is a superclass of $1\#A$ due to $\mathcal{R}$.

Notice that these conditions are a complete description of our approach for detecting the *CAT* pattern. The following example will clarify why such a straightforward approach works.

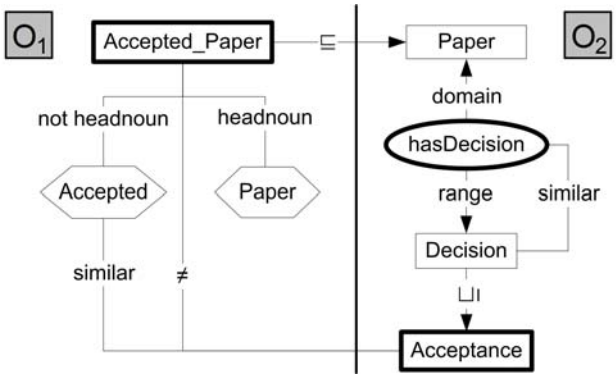


Fig. 2. Conditions relevant for detecting *CAT* correspondence $1\#Accepted\_Paper \equiv \exists 2\#hasDecision.2\#Acceptance$.

With respect to the ontologies depicted in Figure 2 our approach will detect that $1\#Accepted\_Paper \equiv \exists 2\#hasDecision.2\#Acceptance$. The label of *Accepted_Paper* can be split up into prefix `Accepted` and head noun `Paper`. On the one hand the string `Accepted` is similar to `Acceptance`, but on the other hand *Accepted_Paper* = *Acceptance* is not contained in $\mathcal{R}$. Object property *hasDecision* accomplishes all conditions required by our algorithm: *Acceptance* has a superclass *Decision* which is the range of *hasDecision* and the labels `Decision` and `hasDecision` are similar. Moreover the domain of *hasDecision* is a superclass of *Accepted_Paper* due $\mathcal{R}$, which contains correspondence $1\#Paper = 2\#Paper$.

Class by Inverse Attribute Type pattern A correspondence $1\#A \equiv 2\#B \sqcap \exists 2\#R^{-1}.\top$ of the CAT^{-1} type is generated if all following conditions hold.

1. The labels of $1\#A$ and $2\#R$ are similar.
2. There exists a concept $2\#B$ which both is a proper subset of the range of $2\#R$
3. and which is, due to the $\mathcal{R}$, a superclass of $1\#A$.

Notice that for the *CAT* pattern we did not demand similarity between $1\#A$ and $2\#R$. This is related to the fact that the label of a property often describes some aspects of its range and not its domain (e.g. *hasAuthor* relates a paper to its author). Thus, the label of a property is relevant for the inverse pattern CAT^{-1}. The other two conditions are related to structural aspects and filter out candidates that are caused by accidental string similarities.

Class by Attribute Value pattern Although above we described the pattern *CAV* in general, our algorithm will only detect the boolean variant of this pattern. A correspondence $1\#A \equiv \exists 2\#R.\{true\}$ is generated by our algorithm, if all following conditions hold.

1. The range of the datatype property $2\#R$ is `Boolean`.
2. In the following the label of $1\#A$ is split into its head noun $\texttt{hn}(1\#A)$ and the remaining part of the label $\neg\texttt{hn}(1\#A)$. Again, $\neg\texttt{hn}(1\#A)$ is split into a first part $\neg\texttt{hn}_1(1\#A)$ and a remaining part $\neg\texttt{hn}_2(1\#A)$.
   (a) $\texttt{hn}(1\#A)$ is similar to the label of $2\#R$'s domain.
   (b) $\neg\texttt{hn}(1\#A)$ is similar to the label of $2\#R$.
   (c) $\neg\texttt{hn}_1(1\#A)$ is similar to the label of $2\#R$.
3. The domain of $2\#R$ is a superclass of $1\#A$ due to $\mathcal{R}$.

Given a non-boolean datatype property range, more sophisticated techniques are required to decide which set of values is adequate for which concept. In our case this distinction is based on condition 2c. If the similarity value does not exceed a certain threshold, we generate $1\#A \equiv \exists 2\#R.\{false\}$ instead of $1\#A \equiv \exists 2\#R.\{true\}$. An example detected in our experimental study is $1\#Early\_Registered\_Participant \equiv \exists 2\#earlyRegistration.\{true\}$ exploiting $1\#Participant \equiv 2\#Participant$ in $\mathcal{R}$.

Property Chain pattern A correspondence $1\#R \equiv 2\#P \circ 2\#Q$ of type *PC* is generated, if all following conditions hold.

1. Due to $\mathcal{R}$, the domain of $1\#R$ is a subclass or superclass of the domain of $2\#P$.
2. The range of $2\#P$ is a subclass or superclass of the domain of $2\#Q$.
3. Datatype properties $1\#R$ and $2\#Q$ have a compatible data range.
4. The labels of $1\#R$ and $2\#P$ are similar.
5. The label of $2\#Q$ is `name` or is contained in the label of $1\#R$ resp. vice versa.

Due to the condition that range of $2\#P$ and domain of $2\#Q$ are in a superclass relation, the successive application of the properties can be ensured. Often $1\#R$ maps a class onto a name, therefore especially properties which are labeled with `name` are potential

mapping candidates. An example for this pattern has already been given in the previous section. With respect to Figure 1 we have $1\#R$ = $1\#author$, $2\#P$ = $2\#hasAuthor$, $2\#Q$ = $2\#name$. The property $1\#author$ relates a paper to the name of its author, $2\#hasAuthor$ relates a paper to its author and $2\#name$ an author to its name. Thus, a chain of properties is required to express $1\#author$ in the terminology defined by $\mathcal{O}_2$.

A second set of conditions aims to cover a different naming strategy. The first three conditions are the same as above, but the last ones have to be replaced as follows.

4. The labels of $1\#R$ and $2\#Q$ are similar.
5. The labels of $2\#P$ and its range or the labels of the properties $2\#P$ and $2\#Q$ are similar.

An example, depicted in Figure 4, of a property chain that fulfills these conditions: $1\#hasYear$ = $2\#date \circ 2\#year$ where $2\#date$ is an object property with $2\#Date$ as abstract range.

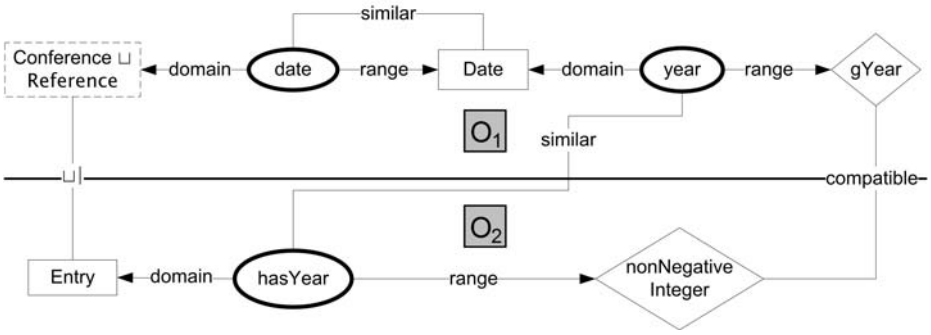


Fig. 3. Conditions relevant for detecting *PC* correspondence $1\#hasYear \equiv 2\#date \circ 2\#year$

For all patterns of the *class by* and *property chain* family we additionally check for each candidate correspondence whether there exists a constituent that already occurs in the reference alignment. In this case we trust the simple correspondence in the reference alignment and do not generate the complex correspondence.

5 Experiments

The algorithms described in the previous section have been implemented in a matching tool available at `http://dominique-ritze.de/complex-mappings/`. We applied our tool on three datasets referred to as CONFERENCE 1, CONFERENCE 2 and BENCHMARK. These datasets have been taken from corresponding tracks of the Ontology Alignment Evaluation Initiative (OAEI). As BENCHMARK we refer to the matching tasks #301 - #304 of the OAEI Benchmark track. We abstained from using the other test cases, because they are generated by systematic variations of the #101 ontology, which

do not exceed a certain degree of structural difference. The CONFERENCE 1 dataset consists of all pairs of ontologies for which a reference alignment is available. Additionally, we used the reference alignment between concepts created for the experiments conducted in [5] to extend our datasets. This dataset is referred to as CONFERENCE 2 and has not been regarded while looking for complex correspondences.
Notice that all conditions in our algorithms express hard boolean constraints. The only exception is the threshold that determines whether two strings are similar. Therefore, we conducted our experiments with different thresholds from 0.6 to 0.9.

	Correct Correspondences (true positives)												Incorrect Correspondences (false positives)											
Type	*CAT* & CAT^{-1}				PC				$\sum$				*CAT* & CAT^{-1}				PC				$\sum$			
Threshold	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9
CONFERENCE 1	7	5	5	0	1	1	1	1	8	6	6	1	16	8	6	2	5	3	2	1	21	11	8	3
CONFERENCE 2	3	3	2	0	0	0	0	0	3	3	2	0	8	6	5	0	14	11	11	7	22	17	16	7
BENCHMARK	0	0	0	0	17	17	17	17	17	17	17	17	0	0	0	0	2	2	1	0	2	2	1	0
$\sum$	**10**	**8**	**7**	**0**	**18**	**18**	**18**	**18**	**28**	**26**	**25**	**18**	**24**	**14**	**11**	**2**	**21**	**16**	**14**	**8**	**45**	**30**	**25**	**10**

Table 1. Results with four different thresholds

Table 1 gives an overview on the results of our experiments. We carefully analyzed all generated correspondences and divided them in correct (true positives) and incorrect ones (false positives). One might first notice that we did not include a column for the *CAV* pattern. Unfortunately, only two correct and one incorrect correspondence of this type have been detected in the CONFERENCE 1 dataset. Remember that we only focused on boolean datatype properties. A more general strategy might result in higher recall. Nevertheless, to our knowledge all correspondences of the boolean *CAV* have been detected and even with low thresholds only one incorrect correspondence accrued.

Obviously there is a clear distinction between different datasets. While our matching system detected correct complex correspondences of *class by* types in the CONFERENCE datasets, none have been detected in the BENCHMARK dataset. Nearly the same holds vice versa. This is based on the fact that the ontologies of the BENCHMARK dataset are dedicated to the very narrow domain of bibliography and do not strongly vary with respect to their concept hierarchy, while differences can be found with regard to the use of properties. The CONFERENCE ontologies on the other hand have very different conceptual hierarchies.

Correspondences of the pattern *CAT* and CAT^{-1} can be found in both CONFERENCE 1 & 2 datasets. As expected we find the typical relation between precision and recall on the one hand and the chosen threshold on the other hand: low thresholds cause low precision of approx 30% and allow to detect a relatively high number of correct correspondences. A nearly balanced ratio between true and false positives is reached with a threshold of 0.8.

For the *PC* pattern a threshold of 0.6 results in 18 correct and 21 incorrect correspondences. Surprisingly, the number of correct correspondences does not decrease with increasing threshold, although the number of incorrect correspondences decreases

significantly. This is based on the fact that the relevant entities occurring in the *PC* pattern are very often not only similar but identical after normalization (e.g. concept *Date* and property *date*). This observation indicates that there is still room for improvement by choosing different thresholds for different patterns.

Another surprising result is the high number of false property chains in the CONFERENCE 1 and in particular in the CONFERENCE 2 dataset compared to the BENCHMARK dataset. Due to the existence of a reference alignment with high coverage of properties for the BENCHMARK dataset many incorrect property chains have not been generated. Their constituents already occurred in simple correspondence of the reference alignment. The same does not hold for the CONFERENCE datasets. There are many properties that have no counterpart in one of the other ontologies.

Our experimental study points to the problem of evaluating the quality of a complex alignment. Due to the fact that complex correspondences are missing in the reference alignments, our results cannot be compared against a gold standard, resulting in missing recall values. Even though it might be possible to construct a complete reference alignment for a finite number of patterns, it will be extremely laborious to construct a complete reference alignment, which contains all non-trivial complex correspondences. Nevertheless, a comparison against the size of the simple reference alignments might deliver some useful insights. The number of property correspondences in the union of all BENCHMARK reference alignments is 139 (only 63 concept correspondences), while we could find 17 additional property chains with our approach. For the CONFERENCE datasets we counted 275 concept correspondences (only the CONFERENCE 1 dataset comprised additionally 12 property correspondences). Here we detected 12 complex correspondences of different class by types. These results indicate that the proposed complex ontology matching strategy increased recall by approx. 4% with respect to concept correspondences and by approx. 10% with repect to property correspondences.

Interpreting these results, we have to keep in mind that the generation of complex correspondences is much harder compared to the generation of simple correspondences. While a balanced rate of correct and incorrect correspondences will not be acceptable for simple matching tasks, a similar result is positive with respect to the complex matching task which we tackle with our approach.

6 Conclusion

We proposed a pattern based approach to detect different types of complex correspondences. Our approach does not rely on machine learning techniques, which require the availability of instance correspondences. On the contrary, it is based on state of the art matching techniques and additionally exploits an input alignment which consists of simple correspondences. In an experimental study we have shown that our approach, which is simply based on checking conditions specific to a particular pattern, is sufficient to detect a significant amount of complex correspondences, while the number of false positives is relatively low, if considering that complex correspondences are quite hard to detect.

Although first results are promising, we know that the task of verifying the correctness of complex correspondences requires human interaction. A pattern based approach,

as proposed in this paper, will in most cases fail to generate highly precise alignments. This is based on the fact that the generation of complex correspondences is significantly harder compared to the task of generating simple correspondences. Suppose, given concept *AcceptedPaper* of $\mathcal{O}_1$, a user is searching in $\mathcal{O}_2$ for an equivalent concept. First of all, there are as much simple hypotheses available as there are atomic concepts in $\mathcal{O}_2$. The situation changes dramatically when there exists no atomic counterpart and a complex correspondence is required. The search space explodes and it becomes impossible for a human expert to evaluate each possible combination. We know that the proposed patterns covers only a small part of an infinite search space. Nevertheless, this small part might still be large enough to find a significant fraction of those correspondences that will not be detected at all without a supporting system.

Acknowledgment The work has been partially supported by the German Science Foundation (DFG) under contract STU 266/3-1 and STU 266/5-1 and by the IGA VSE grant no. 20/08 "Evaluation and matching ontologies via patterns".

References

1. A. Doan and A. Y. Halevy. Semantic-integration research in the database community. *AI Magazine*, pages 83–94, 2005.
2. J. Euzenat, F. Scharffe, and A. Zimmermann. Expressive alignment language and implementation. deliverable 2.2.10, Knowledge web, 2007.
3. J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer, 2007.
4. V. I. Levenshtein. Binary codes capable of correcting deletions and insertions and reversals. *Doklady Akademii Nauk SSSR*, pages 845–848, 1965. In Russian. English Translation in Soviet Physics Doklady, 10(8) p. 707710, 1966.
5. C. Meilicke, A. Tamilin, and H. Stuckenschmidt. Repairing Ontology Mappings. In *Proceedings of the 22nd Conference on Artificial Intelligence*, Vancouver, Canada, 2007.
6. H. Qin, D. Dou, and P. LePendu. Discovering Executable Semantic Mappings Between Ontologies. *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, pages 832–849, 2007.
7. D. Ritze. Generating Complex Ontology Alignments, University Mannheim (Bachelor thesis), 2009.
8. F. Scharffe. *Correspondence Patterns Representation*. PhD thesis, University of Innsbruck, 2009.
9. H. Stuckenschmidt, L. Predoiu, and C. Meilicke. Learning Complex Ontology Alignments A Challenge for ILP Research. In *Proceedings of the 18th International Conference on Inductive Logic Programming*, 2008.
10. O. Šváb, V. Svátek, P. Berka, D. Rak, and P. Tomášek. OntoFarm: Towards an Experimental Collection of Parallel Ontologies. In *Poster Proceedings of the International Semantic Web Conference*, 2005.
11. O. Šváb-Zamazal and V. Svátek. Towards Ontology Matching via Pattern-Based Detection of Semantic Structures in OWL Ontologies. In *Proceedings of the Znalosti Czecho-Slovak Knowledge Technology conference*, 2009.
12. O. Šváb-Zamazal, V. Svátek, J. David, and F. Scharffe. Towards Metamorphic Semantic Models. In *Poster session at European Semantic Web Conference*, 2009.
13. O. Šváb-Zamazal, V. Svátek, and F. Scharffe. Pattern-based Ontology Transformation Service. In *Proceedings of the 1st International Conference on Knowledge Engineering and Ontology Development*, 2009.

Computing minimal mappings

Fausto Giunchiglia, Vincenzo Maltese, Aliaksandr Autayeu

Dipartimento di Ingegneria e Scienza dell'Informazione (DISI) - Università di Trento
{fausto, maltese, autayeu}@disi.unitn.it

Abstract. Given two classifications, or lightweight ontologies, we compute the minimal mapping, namely the subset of all possible correspondences, called mapping elements, between them such that i) all the others can be computed from them in time linear in the size of the input ontologies, and ii) none of them can be dropped without losing property i). In this paper we provide a formal definition of minimal mappings and define a time efficient computation algorithm which minimizes the number of comparisons between the nodes of the two input ontologies. The experimental results show a substantial improvement both in the computation time and in the number of mapping elements which need to be handled.

Keywords: Ontology matching, lightweight ontologies, minimal mappings

1 Introduction

Given any two graph-like structures, e.g., database and XML schemas, classifications, thesauri and ontologies, matching is usually identified as the problem of finding those nodes in the two structures which semantically correspond to one another. Any such pair of nodes, along with the semantic relationship holding between the two, is what we informally call a *mapping element*. In the last few years a lot of work has been done on this topic both in the digital libraries [15, 16, 17, 21] and the computer science [2, 3, 4, 5, 6, 8, 9] communities. In this paper we concentrate on lightweight ontologies (or formal classifications), as formally defined in [1, 7], and we focus on the problem of finding *minimal mappings*, that is, the subset of all possible correspondences, called *mapping elements*, such that i) all the others can be computed from them in time linear in the size of the input graphs, and ii) none of them can be dropped without losing property i). This must not be seen as a limitation. There are plenty of schemas in the world which can be translated, with almost no loss of information, into lightweight ontologies. For instance, thesauri, library classifications, file systems, email folder structures, web directories, business catalogues and so on. Lightweight ontologies are well defined and pervasive. The main advantage of minimal mappings is that they are the minimal amount of information that needs to be dealt with. Notice that this is a rather important feature as the number of possible mapping elements can grow up to $n*m$ with n and m being the size of the two input ontologies. Minimal mappings provide clear usability advantages. Many systems and corresponding interfaces, mostly graphical, have been provided for the management of mappings but all of them hardly scale with the increasing number of nodes, and the resulting visualizations are rather messy [3]. Furthermore, the maintenance of smaller sets makes the work of the user much easier, faster and less error prone [11].

The main contributions of this paper are a formal definition of *minimal* and, dually, *redundant mappings*, evidence of the fact that the minimal mapping always exists and it is unique and an algorithm for computing it. This algorithm has the following main features:

1. It can be proved to be correct and complete, in the sense that it always computes the minimal mapping;
2. It minimizes the number of calls to the node matching function which computes the relation between two nodes. Notice that node matching in the general case amounts to logical reasoning [5], and it may require exponential time;
3. It computes the mapping of maximum size (including the maximum number of redundant elements) as it maximally exploits the information codified in the graph of the lightweight ontologies in input. This, in turn, avoids missing mapping elements due to pitfalls in the node matching functions**,** e.g. because of missing background knowledge [8].

As far as we know very little work has been done on the issue of computing minimal mappings. In general the computation of minimal mappings can be seen as a specific instance of the mapping inference problem [4]. Closer to our work, in [9, 10, 11] the authors use Distributed Description Logics (DDL) [12] to represent and reason about existing ontology mappings. They introduce a few debugging heuristics which remove mapping elements which are redundant or generate inconsistencies from a given set [10]. The main problem of this approach, as also recognized by the authors, is the complexity of DDL reasoning [11]. In our approach, instead of pruning redundant elements, we directly compute the minimal set. Among other things, our approach allows us to minimize the number of calls to node matching.

The rest of the paper is organized as follows. Section 2 provides a motivating example. Section 3 provides the definition for redundant and minimal mappings, and it shows that the minimal set always exists and it is unique. Section 4 describes the algorithm while Section 5 evaluates it. Finally, Section 6 draws some conclusions and outlines the future work.

2 A motivating example

Classifications are perhaps the most natural tool humans use to organize information content. Information items are hierarchically arranged under topic nodes moving from general ones to more specific ones as long as we go deeper in the hierarchy.

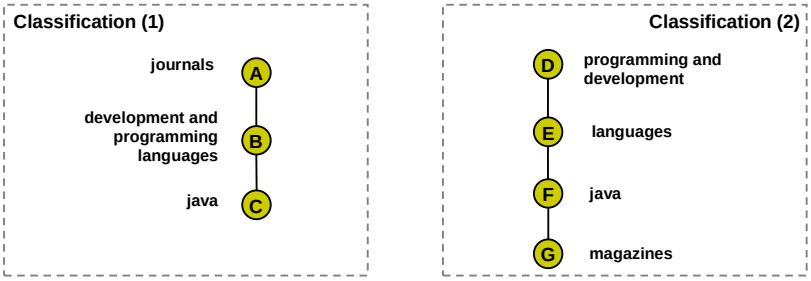
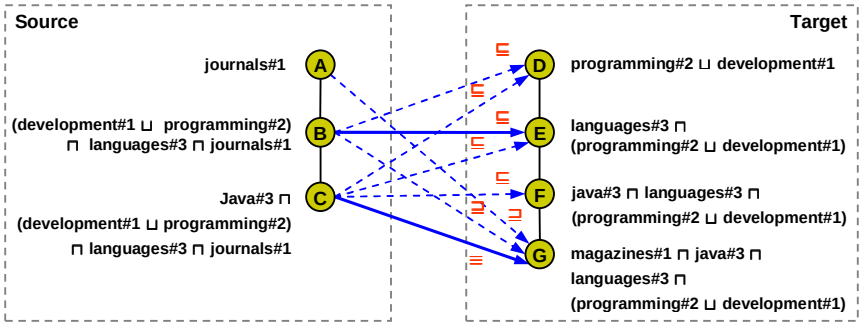


Fig. 1. Two classifications

This attitude is well known in Knowledge Organization as the principle of organizing from the general to the specific [16], called synthetically the *get-specific principle* in [1, 7]. Consider the two fragments of classifications depicted in Fig. 1. They are designed to arrange more or less the same content, but from different perspectives. The second is a fragment taken from the Yahoo web directory[1] (category Computers and Internet).

Following the approach described in [1] and exploiting dedicated NLP techniques tuned to short phrases (for instance, as described in [13]), classifications can be converted, exactly or with a certain degree of approximation, into their formal alter-ego, namely into lightweight ontologies. Lightweight ontologies [1, 7] are acyclic graph structures where each natural language node label is translated into a propositional Description Logic (DL) formula codifying the meaning of the node. Notice that the formula associated to each node contains the formula of the node above to capture the fact that the meaning of each node is contextualized by the meaning of its ancestor nodes. As a consequence, the backbone structure of the resulting lightweight ontologies is represented by subsumption relations between nodes. The resulting formulas are reported in Fig. 2.



M' = {<A, G, ⊒>, <B, D, ⊑>, <B, E, ⊑>, <B, G, ⊒>, <C, D, ⊑>, <C, E, ⊑>, <C, F, ⊑>, <C, G, ≡>}
M = { <B, E, ⊑>, <C, G, ≡>}

Fig. 2. The minimal and redundant mapping between two lightweight ontologies

Here each string denotes a concept (e.g., journals#1) and the number at the end of the strings denote a specific concept constructed from a WordNet sense. Fig. 2 also reports the resulting mapping elements. We assume that each mapping element is associated with one of the following semantic relations: disjointness (⊥), equivalence (≡), more specific (⊑) and less specific (⊒), as computed for instance by semantic matching [5]. Notice however that not all the mapping elements have the same semantic valence. For instance, B⊑D is a trivial logical consequence of B⊑E and E⊑D, and similarly for C⊑F and C≡G. We represent the elements in the minimal mapping using solid lines and redundant elements using dashed lines. M' is the set of maximum size (including the maximum number of redundant elements) while M is the minimal set. The problem is how to compute the minimal set in the most efficent way.

[1]http://dir.yahoo.com/

3 Redundant and minimal mappings

Adapting the definition in [1] we define a lightweight ontology as follows:

Definition 1 (Lightweight ontology). A lightweight ontology O is a rooted tree $<N,E,L^F>$ where:

a) N is a finite set of nodes;
b) E is a set of edges on N;
c) L^F is a finite set of labels expressed in a Propositional DL language such that for any node $n_i \in N$, there is one and only one label $l_i^F \in L^F$;
d) $l_{i+1}^F \sqsubseteq l_i^F$ with n_i being the parent of n_{i+1}.

The superscript F is used to emphasize that labels are in a formal language. Fig. 2 above provides an example of (a fragment of) two lightweight ontologies.

We then define mapping elements as follows:

Definition 2 (Mapping element). Given two lightweight ontologies O_1 and O_2, a mapping element m between them is a triple $<n_1, n_2, R>$, where:

a) $n_1 \in N_1$ is a node in O_1, called the source node;
b) $n_2 \in N_2$ is a node in O_2, called the target node;
c) $R \in \{\equiv, \sqsubseteq, \sqsupseteq, \perp\}$ is the strongest semantic relation holding between n_1 and n_2.

The partial order is such that disjointness is stronger than equivalence which, in turn, is stronger than subsumption (in both directions), and such that the two subsumption symbols are unordered. This is in order to return subsumption only when equivalence does not hold or one of the two nodes being inconsistent (this latter case generating at the same time both a disjointness and a subsumption relation), and similarly for the order between disjointness and equivalence. Notice that, under this ordering, there can be at most one mapping element between two nodes.

The next step is to define the notion of redundancy. The key idea is that, given a mapping element $<n_1, n_2, R>$, a new mapping element $<n_1', n_2', R'>$ is redundant with respect to the first if the existence of the second can be asserted simply by looking at the relative positions of n_1 with n_1', and n_2 with n_2'. In algorithmic terms, this means that the second can be computed without running the time expensive node matching functions. We have identified four basic redundancy patterns as follows:

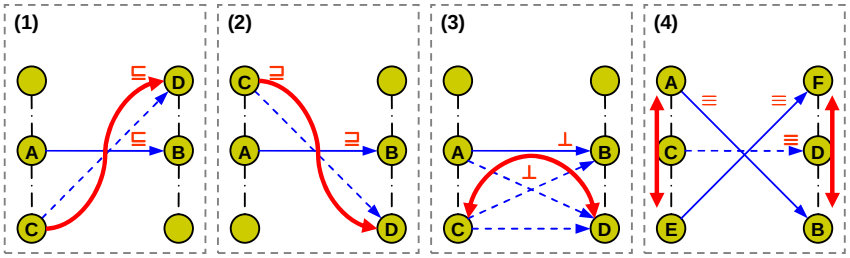


Fig. 3. Redundancy detection patterns

In Fig. 3, the blue dashed mappings are redundant w.r.t. the solid blue ones. The bold red solid lines show how a semantic relation propagates. Let us discuss the rationale for each of the patterns:

- **Pattern (1)**: each mapping element <C, D, ⊑> is redundant w.r.t. <A, B, ⊑>. In fact, C is more specific than A which is more specific than B which is more specific than D. As a consequence, by transitivity C is more specific than D.
- **Pattern (2)**: dual argument as in pattern (1).
- **Pattern (3)**: each mapping element <C, D, ⊥> is redundant w.r.t. <A, B, ⊥>. In fact, we know that A and B are disjoint, that C is more specific than A and that D is more specific than B. This implies that C and D are also disjoint.
- **Pattern (4)**: Pattern 4 is the combinations of patterns (1) and (2).

In other words, the patterns are the way to capture logical inference from structural information, namely just by looking at the position of the nodes in the two trees. As we will show, this on turn allows computing the redundant elements in linear time (w.r.t. the size of the two ontologies) from the ones in the minimal set. Notice that patterns (1) and (2) are still valid in case we substitute subsumption with equivalence. However, in this case we cannot exclude the possibility that a stronger relation holds between C and D. A trivial example of where this is not the case is provided in Fig. 4 (a).

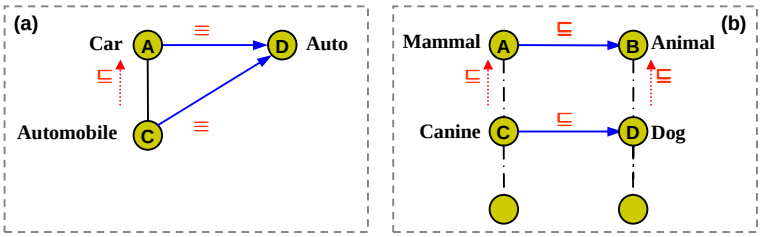


Fig. 4. Examples of non redundant mapping elements

On the basis of the patterns and the considerations above we can define redundant elements as follows. Here path(n) is the path from the root to the node n.

Definition 3 (Redundant mapping element). Given two lightweight ontologies O_1 and O_2, a mapping M and a mapping element m'∈M with m' = <C, D, R'> between them, we say that m' is redundant in M iff one of the following holds:

(1) If R' is ⊑, ∃m∈M with m = <A, B, R> and m ≠ m' such that R ∈ {⊑, ≡}, A ∈ path(C) and D ∈ path(B);

(2) If R' is ⊒, ∃m∈M with m = <A, B, R> and m ≠ m' such that R ∈ {⊒, ≡}, C ∈ path(A) and B ∈ path(D);

(3) If R' is ⊥, ∃m∈M with m = <A, B, ⊥> and m ≠ m' such that A ∈ path(C) and B ∈ path(D);

(4) If R' is ≡, conditions (1) and (2) must be satisfied.

See how Definition 3 maps to the four patterns in Fig. 3. Fig. 2 in Section 2 provides examples of redundant elements. Definition 3 can be proved to capture all and only the cases of redundancy.

Theorem 1 (Redundancy, soundness and completeness). Given a mapping M between two lightweight ontologies O_1 and O_2, a mapping element m' $\in$ M is redundant if and only if it satisfies one of the conditions of Definition 3.

The soundness argument is the rationale described for the patterns above. Completeness can be shown by constructing the counterargument that we cannot have redundancy in the remaining cases. We can proceed by enumeration, negating each of the patterns, encoded one by one in the conditions appearing in the Definition 3. The complete proof is given in [22]. Fig. 4 (b) provides an example of non redundancy which is based on pattern (1). It tells us that the existence of a link between two nodes does not necessarily propagate to the two nodes below. For example we cannot derive that Canine $\sqsubseteq$ Dog from the set of axioms {Canine $\sqsubseteq$ Mammal, Mammal $\sqsubseteq$ Animal, Dog $\sqsubseteq$ Animal}, and it would be wrong to do so.

The notion of redundancy allows us to formalize the notion of minimal mapping as follows:

Definition 4 (Minimal mapping). Given two lightweight ontologies O_1 and O_2, we say that a mapping M between them is minimal iff:

a) $\nexists$m$\in$M such that m is redundant (minimality condition);

b) $\nexists$M'$\supset$M satisfying condition a) above (maximality condition).

A mapping element is minimal if it belongs to the minimal mapping.

Note that conditions (a) and (b) ensure that the minimal set is the set of maximum size with no redundant elements. As an example, the set M in Fig. 2 is minimal. Comparing this mapping with M' we can observe that all elements in the set M' - M are redundant and that, therefore, there are no other supersets of M with the same properties. In effect, <A, G, $\sqsupseteq$> and <B, G, $\sqsupseteq$> are redundant w.r.t. <C, G, $\equiv$> for pattern (2); <C, D, $\sqsubseteq$>, <C, E, $\sqsubseteq$> and <C, F, $\sqsubseteq$> are redundant w.r.t. <C, G, $\equiv$> for pattern (1); <B, D, $\sqsubseteq$> is redundant w.r.t. <B, E, $\sqsubseteq$> for pattern (1). Note that M contains far less mapping elements w.r.t. M'.

As last observation, for any two given lightweight ontologies, the minimal mapping always exists and it is unique.

Theorem 2 (Minimal mapping, existence and uniqueness). Given two lightweight ontologies O_1 and O_2, there is always one and only one minimal mapping between them.

A proof is given in [22].

4 Computing minimal and redundant mappings

The patterns described in the previous section suggest how to significantly reduce the amount of calls to the node matchers. By looking for instance at pattern (2) in Fig. 3, given a mapping element m = <A, B, ⊒> we know that it is not necessary to compute the semantic relation holding between A and any descendant C in the sub-tree of B since we know in advance that it is ⊒. At the top level the algorithm is organized as follows:

- **Step 1, computing the minimal mapping modulo equivalence:** compute the set of disjointness and subsumption mapping elements which are *minimal modulo equivalence*. By this we mean that they are minimal modulo collapsing, whenever possible, two subsumption relations of opposite direction into a single equivalence mapping element;
- **Step 2, computing the minimal mapping:** eliminate the redundant subsumption mapping elements. In particular, collapse all the pairs of subsumption elements (of opposite direction) between the same two nodes into a single equivalence element. This will result into the *minimal mapping*;
- **Step 3, computing the mapping of maximum size:** Compute the mapping of maximum size (including minimal and redundant mapping elements). During this step the existence of a (redundant) element is computed as the result of the propagation of the elements in the minimal mapping.

The first two steps are performed at matching time, while the third is activated whenever the user wants to exploit the pre-computed mapping elements, for instance for their visualization. For lack of space in the following we give only the pseudo-code for the first step. The interested reader can look at [22] for the pseudo-code of the other two steps.

The minimal mapping is computed by a function **TreeMatch** whose pseudo-code is given in Fig. 5. M is the minimal set while T1 and T2 are the input lightweight ontologies.

```
10  node: struct of {cnode: wff; children: node[];}
20  T1,T2: tree of (node);
30  relation in {⊑, ⊒, ≡, ⊥};
40  element: struct of {source: node; target: node; rel: relation;};
50  M: list of (element);
60  boolean direction;

70  function TreeMatch(tree T1, tree T2)
80   {TreeDisjoint(root(T1),root(T2));
90    direction := true;
100   TreeSubsumedBy(root(T1),root(T2));
110   direction := false;
120   TreeSubsumedBy(root(T2),root(T1));
130   TreeEquiv();
140  };
```

Fig. 5. Pseudo-code for the tree matching function

TreeMatch is crucially dependent on the node matching functions **NodeDisjoint** (given in [22]) and **NodeSubsumedBy** (Fig. 6) which take two nodes n1 and n2 and

return a positive answer respectively in case of disjointness or subsumption, or a negative answer if it is not the case or they are not able to establish it. Notice that these two functions hide the heaviest computational costs; in particular their computation time is exponential when the relation holds, but possibly much faster, when the relation does not hold. The main motivation for this is that the node matching problem, in the general case, should be translated into disjointness or subsumption problem in propositional DL (see [5] for a detailed description). The goal, therefore, is to compute the minimal mapping by minimizing the calls to the node matching functions and, in particular minimizing the calls where the relation will turn out to hold. We achieve this purpose by processing both trees top down. To maximize the performance of the system, **TreeMatch** has therefore been built as the sequence of three function calls: the first call to **TreeDisjoint** (line 80) computes the minimal set of disjointness mapping elements, while the second and the third call to **TreeSubsumedBy** compute the minimal set of subsumption mapping elements in the two directions modulo equivalence (lines 90-120). Notice that in the second call, **TreeSubsumedBy** is called with the input ontologies with swapped roles. These three calls correspond to Step 1 above. Line 130 in the pseudo code of the **TreeMatch** implements the Step 2.

Given two sub-trees in input, rooted in n1 and n2, the **TreeDisjoint** function searches for the first disjointness elements along any pair of paths in them. Look at [22] for corresponding pseudo-code and the complete description.

TreeSubsumedBy (Fig. 6) recursively finds all minimal mapping elements where the strongest relation between the nodes is $\sqsubseteq$ (or dually, $\sqsupseteq$ in the second call in the **TreeMatch**, line 120. In the following we will concentrate only on the first call).

```
10  function boolean TreeSubsumedBy(node n1, node n2)
20   {c1,c2: node; LastNodeFound: boolean;
30    if (<n1,n2,⊥> ∈ M) then return false;
40    if (!NodeSubsumedBy(n1, n2)) then
50      foreach c1 in GetChildren(n1) do TreeSubsumedBy(c1,n2);
60    else
70      {LastNodeFound := false;
80       foreach c2 in GetChildren(n2) do
90         if (TreeSubsumedBy(n1,c2)) then LastNodeFound := true;
100      if (!LastNodeFound) then AddSubsumptionMappingElement(n1,n2);
120      return true;
140     };
150   return false;
160  };
170 function boolean NodeSubsumedBy(node n1, node n2)
180  {if (Unsatisfiable(mkConjunction(n1.cnode,negate(n2.cnode)))) then
        return true;
190   else return false; };
200 function AddSubsumptionMappingElement(node n1, node n2)
210  {if (direction) then AddMappingElement(<n1,n2,⊑>);
220   else AddMappingElement(<n2,n1,⊒>); };
```

Fig. 6. Pseudo-code for the **TreeSubsumedBy** function

Notice that **TreeSubsumedBy** assumes that the minimal disjointness elements are already computed. As a consequence, at line 30 it checks whether the mapping ele-

ment between the nodes n1 and n2 is already in the minimal set. If this is the case it stops the recursion. This allows computing the stronger disjointness relation rather than subsumption when both hold (namely in presence of an inconsistent node). Given n2, lines 40-50 implement a depth first recursion in the first tree till a subsumption is found. The test for subsumption is performed by the **NodeSubsumedBy** function that checks whether the formula obtained by the conjunction of the formulas associated to the node n1 and the negation of the formula for n2 is unsatisfiable (lines 170-190). Lines 60-140 implement what happens after the first subsumption is found. The key idea is that, after finding the first subsumption, **TreeSubsumedBy** keeps recursing down the second tree till it finds the last subsumption. When this happens, the resulting mapping element is added to the minimal set (line 100). Notice that both **NodeDisjoint** and **NodeSubsumedBy** call the function **Unsatisfiable** which embeds a call to a SAT solver.

To fully understand **TreeSubsumedBy,** the reader should check what happens in the four situations in Fig. 7. In case (a) the first iteration of the **TreeSubsumedBy** finds a subsumption between A and C. Since C has no children, it skips lines 80-90 and directly adds the mapping element <A, C, ⊑> to the minimal set (line 100). In case (b), since there is a child D of C the algorithm iterates on the pair A-D (lines 80-90) finding a subsumption between them. Since there are no other nodes under D, it adds the mapping element <A, D, ⊑> to the minimal set and returns true. Therefore LastNodeFound is set to true (line 90) and the mapping element between the pair A-C is recognized as redundant. Case (c) is similar. The difference is that **TreeSubsumedBy** will return false when checking the pair A-D (line 30), thanks to previous computation of minimal disjointness mapping elements, and therefore the mapping element <A, C, ⊑> is recognized as minimal. In case (d) the algorithm iterates after the second subsumption mapping element is identified. It first checks the pair A-C and iterates on A-D concluding that subsumption does not hold between them (line 40). Therefore, it recursively calls **TreeSubsumedBy** between B and D. In fact, since <A, C, ⊑> will be recognized as minimal, it is not worth checking <B, C, ⊑> for pattern (1). As a consequence <B, D, ⊑> is recognized as minimal together with <A, C, ⊑>.

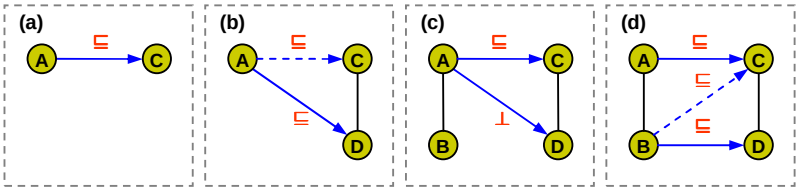


Fig. 7. Examples of applications of the **TreeSubsumedBy**

Five observations. The first is that, even if, overall, **TreeMatch** implements three loops instead of one, the wasted (linear) time is largely counterbalanced by the exponential time saved by avoiding a lot of useless calls to the SAT solver. The second is that, when the input trees T1 and T2 are two nodes, **TreeMatch** behaves as a node matching function which returns the semantic relation holding between the input nodes. The third is that the call to **TreeDisjoint** before the two calls to **TreeSubsumedBy** allows us to implement the partial order on relations defined in the previous

section. In particular it allows returning only a disjointness mapping element when both disjointness and subsumption hold. The fourth is the fact that skipping (in the body of the **TreeDisjoint**) the two sub-trees where disjointness holds is what allows not only implementing the partial order (see the previous observation) but also saving a lot of useless calls to the node matching functions. The fifth and last observation is that the implementation of **TreeMatch** crucially depends on the fact that the minimal elements of the two directions of subsumption and disjointness can be computed independently (modulo inconsistencies).

5 Evaluation

The algorithm presented in the previous section, let us call it MinSMatch, has been implemented by taking the node matching routines of the state of the art matcher SMatch [5] and by changing the way the tree structure is matched. The evaluation has been performed by directly comparing the results of MinSMatch and SMatch on several real-world datasets. All tests have been performed on a Pentium D 3.40GHz with 2GB of RAM running Windows XP SP3 operating system with no additional applications running except the matching system. Both systems were limited to allocating no more than 1GB of RAM. The tuning parameters were set to the default values. The selected datasets had been already used in previous evaluations, see [14]. Some of these datasets can be found at OAEI web site[2]. The first two datasets describe courses and will be called **C**ornell and **W**ashington, respectively. The second two come from the arts domain and will be referred to as **T**opia and **I**con, respectively. The third two datasets have been extracted from the Looksmart, Google and Yahoo! directories and will be referred to as **S**ource and **T**arget. The fourth two datasets contain portions of the two business directories eCl@ss[3] and UNSPSC[4] and will be referred to as **E**class and **U**nspsc. Table 1 describes some indicators of the complexity of these datasets.

#	Dataset pair	Node count	Max depth	Average branching factor
1	Cornell/Washington	34/39	3/3	5.50/4.75
2	Topia/Icon	542/999	2/9	8.19/3.66
3	Source/Target	2857/6628	11/15	2.04/1.94
4	Eclass/Unspsc	3358/5293	4/4	3.18/9.09

Table 1. Complexity of the datasets

Consider Table 2. The reduction in the last column is calculated as (1-m/t), where m is the number of elements in the minimal set and t is the total number of elements in the mapping of maximum size, as computed by MinSMatch. As it can be easily noticed, we have a significant reduction, in the range 68-96%.

The second interesting observation is that in Table 2, in the last two experiments, the number of total mapping elements computed by MinSMatch is slightly higher (compare the second and the third column). This is due to the fact that in the presence of one of the patterns, MinSMatch directly infers the existence of a mapping element without testing it. This allows MinSMacth, differently from SMatch, to avoid missing

[2] http://oaei.ontologymatching.org/2006/directory/
[3] http://www.eclass-online.com/
[4] http://www.unspsc.org/

elements because of failures of the node matching functions (because of lack of background knowledge [8]). One such example from our experiments is reported below (directories Source and Target):

\Top\Computers\Internet\Broadcasting\Video Shows
\Top\Computing\Internet\Fun & Games\Audio & Video\Movies

We have a minimal mapping element which states that Video Shows ⊒ Movies. The element generated by this minimal one, which is captured by MinSMatch and missed by SMatch (because of the lack of background knowledge about the relation between 'Broadcasting' and 'Movies') states that Broadcasting ⊒ Movies.

	S-Match	MinSMatch		
#	Total mapping elements (t)	Total mapping elements (t)	Minimal mapping elements (m)	Reduction, %
1	223	223	36	83.86
2	5491	5491	243	95.57
3	282638	282648	30956	89.05
4	39590	39818	12754	67.97

Table 2. Mapping sizes.

To conclude our analysis, Table 3 shows the reduction in computation time and calls to SAT. As it can be noticed, the time reductions are substantial, in the range 16% - 59%, but where the smallest savings are for very small ontologies. In principle, the deeper the ontologies the more we should save. The interested reader can refer to [5, 14] for a detailed qualitative and performance evaluation of SMatch w.r.t. other state of the art matching algorithms.

	Run Time, ms			SAT calls		
#	S-Match	MinSMatch	Reduction, %	S-Match	MinSMatch	Reduction, %
1	472	397	15.88	3978	2273	42.86
2	141040	67125	52.40	1624374	616371	62.05
3	3593058	1847252	48.58	56808588	19246095	66.12
4	6440952	2642064	58.98	53321682	17961866	66.31

Table 3. Run time and SAT problems

6 Conclusions

In this paper we have provided a definition and a fast algorithm for the computation of the minimal mapping between two lightweight ontologies. The evaluation shows a substantial improvement in the (much lower) computation time, in the (much lower) number of elements which need to be stored and handled and in the (higher) total number of mapping elements which are computed.

The future work includes the experimentation with various large Knowledge Organization Systems (e.g., NALT, AGROVOC, LCSH).

References

1. F. Giunchiglia, M. Marchese, I. Zaihrayeu, 2006. Encoding Classifications into Lightweight Ontologies. Journal of Data Semantics 8, pp. 57-81.

2. P. Shvaiko, J. Euzenat, 2007. Ontology Matching. Springer-Verlag New York, Inc. Secaucus, NJ, USA.
3. P. Shvaiko, J. Euzenat, 2008. Ten Challenges for Ontology Matching. In Proceedings of the 7th International Conference on Ontologies, Databases, and Applications of Semantics (ODBASE 2008).
4. J. Madhavan, P. A. Bernstein, P. Domingos, A. Y. Halevy, 2002. Representing and Reasoning about Mappings between Domain Models. At the 18th National Conference on Artificial Intelligence (AAAI 2002).
5. F. Giunchiglia, M. Yatskevich, P. Shvaiko, 2007. Semantic Matching: algorithms and implementation. Journal on Data Semantics, IX, 2007.
6. C. Caracciolo, J. Euzenat, L. Hollink, R. Ichise, A. Isaac, V. Malaisé, C. Meilicke, J. Pane, P. Shvaiko, 2008. First results of the Ontology Alignment Evaluation Initiative 2008.
7. F.Giunchiglia, I. Zaihrayeu, 2007. Lightweight Ontologies. In The Encyclopedia of Database Systems, to appear. Springer, 2008.
8. F.Giunchiglia, P. Shvaiko, M. Yatskevich, 2006. Discovering missing background knowledge in ontology matching. Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006), pp. 382–386.
9. H. Stuckenschmidt, L. Serafini, H. Wache, 2006. Reasoning about Ontology Mappings. Proceedings of the ECAI-06 Workshop on Contextual Representation and Reasoning.
10. C. Meilicke, H. Stuckenschmidt, A. Tamilin, 2006. Improving automatically created mappings using logical reasoning. In the proceedings of the 1st International Workshop on Ontology Matching OM-2006, CEUR Workshop Proceedings Vol. 225.
11. C. Meilicke, H. Stuckenschmidt, A. Tamilin, 2008. Reasoning support for mapping revision. Journal of Logic and Computation, 2008.
12. A. Borgida, L. Serafini. Distributed Description Logics: Assimilating Information from Peer Sources. Journal on Data Semantics pp. 153-184.
13. I. Zaihrayeu, L. Sun, F. Giunchiglia, W. Pan, Q. Ju, M. Chi, and X. Huang, 2007. From web directories to ontologies: Natural language processing challenges. In 6th International Semantic Web Conference (ISWC 2007).
14. P. Avesani, F. Giunchiglia and M. Yatskevich, 2005. A Large Scale Taxonomy Mapping Evaluation. In Proceedings of International Semantic Web Conference (ISWC 2005), pp. 67-81.
15. M. L. Zeng, L. M. Chan, 2004. Trends and Issues in Establishing Interoperability Among Knowledge Organization Systems. Journal of the American Society for Information Science and Technology, 55(5) pp. 377–395.
16. L. Kovács. A. Micsik, 2007. Extending Semantic Matching Towards Digital Library Contexts. Proc.eedings of the 11th European Conference on Digital Libraries (ECDL 2007), pp. 285-296.
17. B. Marshall, T. Madhusudan, 2004. Element matching in concept maps. Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2004), pp.186-187.
18. B. Hjørland, 2008. What is Knowledge Organization (KO)?. Knowledge Organization. International Journal devoted to Concept Theory, Classification, Indexing and Knowledge Representation 35(2/3) pp. 86-101.
19. D. Soergel, 1972. A Universal Source Thesaurus as a Classification Generator. Journal of the American Society for Information Science 23(5), pp. 299–305.
20. D. Vizine-Goetz, C. Hickey, A. Houghton, and R. Thompson. 2004. Vocabulary Mapping for Terminology Services. Journal of Digital Information, Volume 4, Issue 4.
21. M. Doerr, 2001. Semantic Problems of Thesaurus Mapping. Journal of Digital Information, Volume 1, Issue 8.
22. F. Giunchiglia, V. Maltese, A. Autayeu, 2008. Computing minimal mappings. University of Trento, DISI Technical Report: http://eprints.biblio.unitn.it/archive/00001525/

Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods*

Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe

ADVIS Lab
Department of Computer Science
University of Illinois at Chicago
{ifc | flav | cstroe1}@cs.uic.edu

Abstract. The AgreementMaker system for ontology matching includes an extensible architecture that facilitates the integration and performance tuning of a variety of matching methods, an evaluation mechanism, which can make use of a reference matching or rely solely on "inherent" quality measures, and a multi-purpose user interface, which drives both the matching methods and the evaluation strategies. In this paper, we focus on two main features of AgreementMaker. The former is an optimized method that performs the selection of mappings given the similarities between entities computed by any matching algorithm, a threshold value, and the desired cardinalities of the mappings. Experiments show that our method is more efficient than the typically adopted combinatorial method. The latter is the evaluation framework, which includes three "inherent" quality measures that can be used both to evaluate matching methods when a reference matching is not available and to combine multiple matching results by defining the weighting scheme of a *fully automatic* combination method.

1 Introduction

The quest for correctness, completeness, and efficiency in the process of finding correspondences (or mappings) between semantically related entities of different real-world ontologies is a difficult and challenging task for several reasons. For example, an algorithm may be effective for a given scenario, but not for others. Even within the same scenario, the use of different parameters can change the outcome significantly. Therefore, state-of-the-art ontology matching systems [8] tend to adopt different strategies within the same infrastructure even though the intelligent combination of multiple matching results is still an open problem.

Our collaboration with domain experts in the geospatial domain [3] has revealed that they value automatic matching methods, especially for ontologies with thousands of concepts. However, they want to be able to evaluate the matching process, thus requiring to be directly involved in the loop. Such considerations have motivated the most recent features of the AgreementMaker system[1] for ontology matching [1, 2]. These features include a comprehensive user interface supporting both advanced visualization techniques and a control panel that drives all the matching methods and evaluation strategies (Figure 1) and an extensible architecture to incorporate new methods easily and to tune their performance. In this paper we concentrate on an optimization technique to produce the final set of mappings efficiently and on

* Research supported by NSF Awards ITR IIS-0326284, IIS-0513553, and IIS-0812258.

[1] www.AgreementMaker.org.

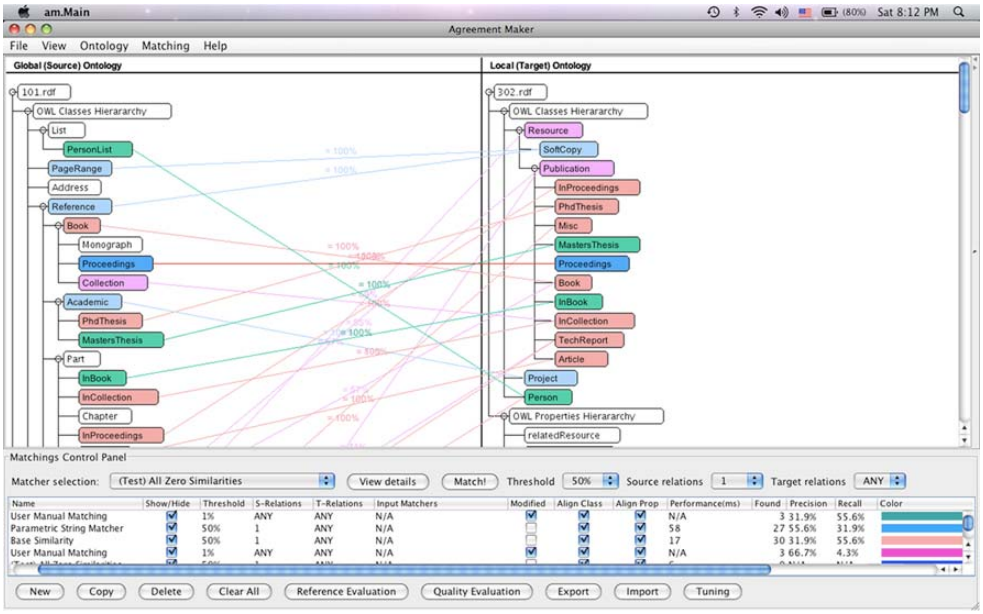


Fig. 1. User interface displaying side-by-side the source and target ontologies (top) and the control panel for the evaluation and comparison of matching methods (bottom).

the system's capability to evaluate, compare, and combine different strategies and matching results.

We describe next the main components of the paper. In Section 2, we cover related work. In Section 3, we describe several of the matching methods, or *matchers*, and their organization in *layers*. The ontologies being matched are called *source* and *target* ontologies. Matchers perform *similarity computation* in which each concept of the source ontology is compared with all the concepts of the target ontology, thus producing two similarity matrices (one for classes and one for properties), which contain a value for each pair of concepts.

In Section 4, we describe the process of *mappings selection* in which a similarity matrix is scanned to select the best mappings according to a given threshold and to the cardinality of the correspondences. For the mappings selection, we distinguish the following four cases: 1-1, n-m, n-$*$ (analogous to $*$-m), $*$-$*$, where 1, n, and m indicate specific input parameters and $*$ indicate that there is no constraint on the number of relations. For example, 1-1 means that each concept in the source ontology will be matched with at most one concept in the target ontology, n-m means that each concept in the source ontology will be matched with at most m concepts in the target ontology, whereas each concept in the target ontology will be matched with at most n concepts in the source ontology. In the case n-$*$, for example, each concept in the source ontology can be matched to any number of concepts in the target ontology. We note that in this case the chosen similarity threshold will in fact determine the number of concepts in the target ontology. In order to maximize the overall similarity of the selected mappings in a 1-1 or n-m matching, an optimization problem (namely the Assignment Problem) has to be solved. We provide an efficient solution to this problem by reducing it to the maximum weight matching in a bipartite graph and by adopting the Shortest Augmenting Path algorithm (SAP) [11]. Our experiments,

which we describe in Section 6, have shown that this solution is considerably more efficient both space- and time-wise than the typically used Hungarian Method [12].

In Section 5, we describe the evaluation framework, which can make use of a reference matching or rely solely on "inherent" quality measures. In particular, we have adopted in our system two quality measures proposed by others [10], namely *order* and *distance preservation*, which analyze the structural properties of the produced matching to help determine its quality, and our own quality measure, called *local confidence*, which measures the reliability of the similarity measures assigned by a matching method. In addition, users can adopt any of these quality measures to define the weighting scheme of a fully automatic method that combines multiple matchings. The experiments, reported in Section 6, have shown that the *local confidence* quality measure can be quite effective in such a task.

2 Related Work

There are several notable systems related to ours [7, 8]. In this section, we will look at related systems with a special focus on the topics of combination of matching methods, mappings selection, and quality measures.

RiMOM [15] implements more than eight different matchers. It adopts a strategy selection method based on the definition of three ontology feature factors: *label similarity*, *structure similarity*, and *label meaning*. These factors are estimated based on the two ontologies to be matched. The matching strategies to be used are those that are suited to the highest factors. For example, if the two ontologies have high label similarity factor, then RiMOM will mostly rely on linguistic based strategies; while if the two ontologies have a high structure similarity factor, it will employ similarity-propagation based strategies on them. However, we note that the association between factors and strategies is predefined. Multiple results are combined using the weighted average of their similarity values, where the weights are predefined experimentally. While AgreementMaker does not provide a strategy selection method, it also provides a combination strategy based on the linear interpolation of the similarity values. However, in contrast with the RiMOM system, the weights can be either user assigned or evaluated through automatically-determined quality measures. This framework is extensible, because if a new method is integrated into the system, it can be directly used and combined with other methods. In terms of the final selection of mappings, RiMOM uses a similarity threshold value, while AgreementMaker uses in addition cardinality values.

Falcon-AO [9] uses four elementary matchers. Similarly to RiMOM, the association between detected similarities and matchers to be combined are predefined. However, matching results can only be combined two at a time (thus differing from both RiMOM and AgreementMaker). While RiMOM does not provide any evaluation strategy, Falcon-AO allows users to evaluate the precision, recall, and F-measure of a matching method given a reference matching. As for the mappings selection phase, Falcon-AO (like RiMOM) does not consider cardinality parameters.

SAMBO and SAMBOdtf [13] have five basic matchers, which are combined using the weighted average of similarities, where the weights are predefined. As for the mappings selection phase, SAMBOdtf adopts a strategy that is based on double threshold: pairs above the threshold are are retained as suggestions, those in between the lower and the upper threshold are filtered using structural information, and the rest is discarded.

None of the above systems proposes quality measures. One approach in this direction reduces mapping incoherence of the computed mappings to concept unsatisfiability in the ontology that results from merging matched ontologies [14]. The quality evaluation is then computed by measuring the effort necessary to remove all causes of incoherence from the matching.

Mapping incoherence is also used in the ILIADS system [16], which performs ontology matching and merging. They start by matching concepts, which are logical mappings that are used to create a unique integrated ontology. Logical reasoning over the constraints in the ontologies creates a consistent integrated ontology.

Other work proposes new measures that extend precision and recall to objects that are semantically defined, such as those in ontologies and alignments [5]. Such quality measures could be integrated into AgreementMaker, in addition to the "classically" defined concepts of precision and recall already supported.

3 Matching Methods

Our architecture allows for serial and parallel composition where, respectively, the output of one or more methods can be used as input to another one, or several methods can be used on the same input and then combined. A set of mappings may therefore be the result of a sequence of steps, called *layers*, to obtain a final *matching* or *alignment* (i.e., a set of mappings).

First layer matchers compare concept features (e.g., label, comments, annotations, and instances) and use a variety of methods including syntactic and lexical comparison algorithms as well as the use of a lexicon like WordNet in the Base Similarity Matcher (BSM) [4]. In the Parametric String-based Matcher (PSM) (see Figure 2), users can choose between a set of string comparison metrics (i.e., edit-distance, Jaro-Winkler, and a substring-based measure devised by us), define the normalization process (e.g., stemming, stop-word removing, and link stripping), and weigh the relevance of each considered concept feature. The similarity between two concepts is computed as the weighted average of the similarities between their single features.

In several methods, the common information between two concepts is kept into separate features and compared within each feature: labels are compared with labels and concept descriptions are compared with concept descriptions, for example. For this reason, we adopt a Vector-based Multi-word Matcher (VMM) that treats concepts as virtual documents containing the information pertaining to them. This information includes their descriptions, the information about their neighbors, and extensional information (e.g., class instances). These containers of terms are transformed into TF-IDF vectors and the similarity is computed using the cosine similarity metric, which is a common technique used to compare documents (see Figure 3).

Second layer matchers use structural properties of the ontologies. Our own methods include the Descendant's Similarity Inheritance (DSI) and the Sibling's Similarity Contribution (SSC) [4]. As their name indicates, they take respectively into account the information about concepts of which a given concept is a descendant or sibling.

Finally, third layer matchers combine the results of two or more matchers so as to obtain a unique final matching in two steps. In the first step, a similarity matrix is built for each pair of concepts, using our Linear Weighted Combination (LWC) matcher, which processes the weighted average for the different similarity results (see Figure 4). Weights can be assigned manually or automatically, the latter kind being determined using our evaluation methods (presented in Section 5). The second

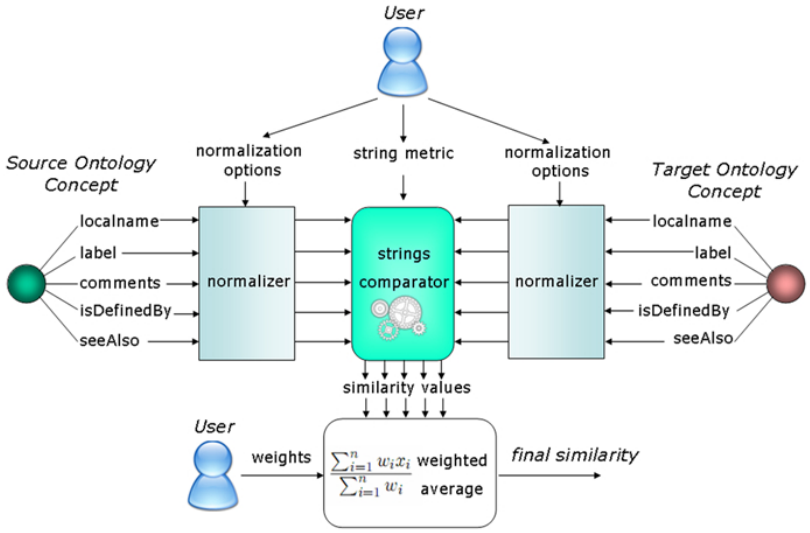


Fig. 2. Parametric String-based Matcher (PSM).

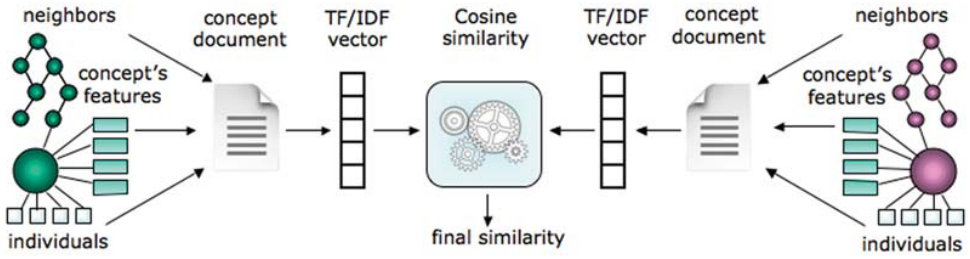


Fig. 3. Vector-based Multi-word Matcher (VMM).

step uses that similarity matrix and takes into account a similarity value and the desired cardinality to generate the final set of mappings, which maximizes the overall similarity while satisfying the selection constraints.

4 Mappings Selection

Four cases are considered in the selection process (see Section 1), depending on the desired cardinality: 1-1, n-m, n-$*$ (analogous to $*$-m), and $*$-$*$. The solution to n-$*$ can be found by scanning each row in the similarity matrix (or each column in the case $*$-m) and by selecting the n most similar correspondences with similarity values higher than the threshold (see Figure 5). For the $*$-$*$ case, only the threshold constraint has to be satisfied.

The 1-1 matching case, which is often required in real-world scenarios, is a challenging problem. In order to maximize the overall similarity in such scenarios, an optimization problem (namely the Assignment Problem) has to be solved. Usually, combinatorial algorithms (e.g., the Hungarian Method [12]) are used to find the optimal solution, but they are costly in terms of space usage and execution time and are for this reason impractical to match ontologies with thousands of concepts.

We provide an efficient alternative solution to this problem by reducing it to the maximum weight matching in the bipartite graph $G = (S \cup T, E)$, where S

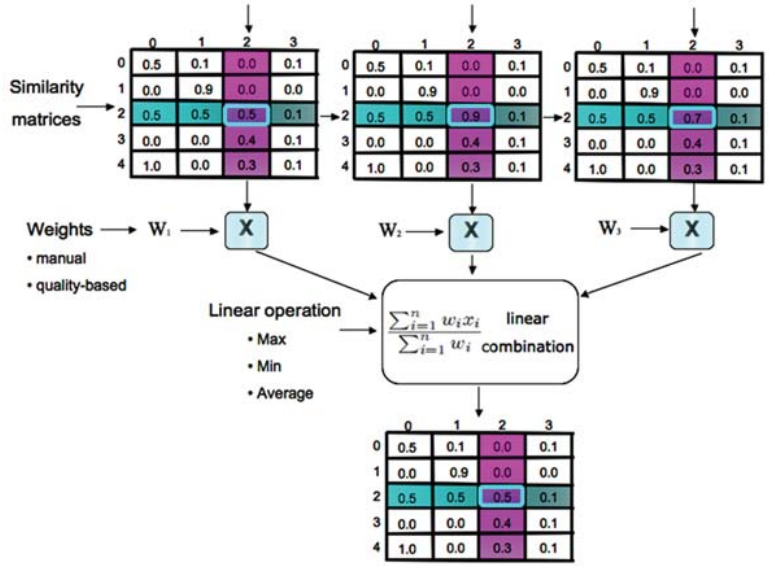


Fig. 4. Linear Weighted Combination (LWC) matcher.

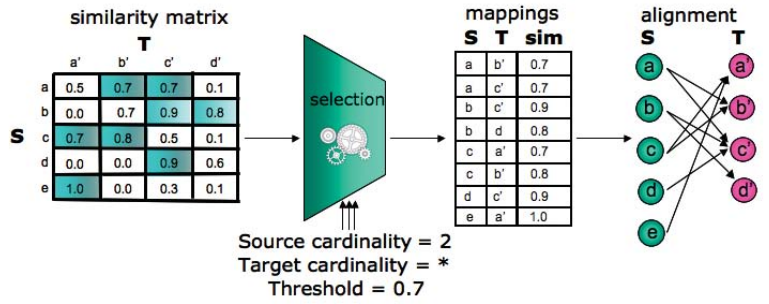


Fig. 5. Example of 2-∗ matching with threshold 0.7 illustrating the similarity level (which yields the similarity matrix) and the selection level (which yields the alignment).

contains the source ontology concepts, T contains the target ontology concepts, and E contains an edge oriented from S to T for each correspondence with a similarity value higher than the threshold, weighted with the threshold value itself. We recall that a maximum weight matching M is a subset of the edges in E such that for each vertex in G at most one adjacent edge is contained in M and the sum of the weights (i.e., the similarity values) of the selected edges is maximized. Thanks to this transformation, we can adopt the Shortest Augmenting Path algorithm (SAP) [11] to find the optimal solution in polynomial time.

Finally, in the n-m selection case, we reuse our algorithm for the 1-1 matching case several times sequentially. We keep track of the number of mappings found for each vertex, and at the end of each iteration, we remove from the bipartite graph all the vertices together with their adjacent edges that have reached the maximal cardinality. The algorithm terminates when the graph is empty. We do not know of any other ontology matching system that has investigated the selection process with this level of detail.

	similarity level	selection level
local	Quality of each row (or column) of the similarity table	Quality of a mapping
global	Quality of the entire similarity table	Quality of the whole set of mappings

Table 1. Categorization of measures of quality of a matching method.

5 Evaluation

The most effective evaluation technique compares the mappings found by the system between the two ontologies with a reference matching or "gold standard," which is a complete set of correct mappings as built by domain experts, in order to measure precision, recall, and F-measure. The AgreementMaker system supports this evaluation technique. In addition, a reference matching can also be used to tune algorithms by using a feedback mechanism provided by a succession of runs.

However, a gold standard is usually not available. Therefore, "inherent" quality measures need to be considered. These measures can be defined at two levels as associated with the two main modules of a matcher: *similarity* or *selection* level. As illustrated in Figure 5, we can consider *local* quality as associated with a single row (or a single column) of the similarity matrix at the *similarity* level (or mapping at the *selection* level) or *global* quality as associated with all the correspondences in the similarity matrix at the *similarity* level (or with all the mappings in a matching at the *selection* level). This categorization of quality measures is summarized in Table 1. We have incorporated in our system two *global-selection* quality measures proposed by others [10] and one *local-similarity* quality measure that we have devised.

The intuition behind the two *global-selection* quality measures, namely (1) *order* and (2) *distance preservation*, is that given a set of mappings we can measure the structural properties of the produced matching to help determine its quality. In particular, according to (1), a matching should not change the order of concepts as defined by the *is-a* or *part-of* relations, and, according to (2), it should preserve the distance between concepts as much as possible. These metrics are good measures of the quality of a set of mappings if the ontologies are structurally similar.

In contrast with the two *global-selection* measures, the *local-similarity* quality measure is independent of the properties of the ontologies. Indeed, it tries to measure the reliability of the similarity measures assigned by a matching method, which is an intrinsic property of the matching method and therefore scenario independent. In particular, for each source (or target) concept we want to measure the confidence of the matcher as related to the selected mappings for that concept. Similarity-based matching techniques are based on the idea that if two concepts are very similar, they probably deserve to be matched. Therefore, our measure should be directly proportional to the similarity values of selected mappings. At the same time, we want to detect and penalize those matchers that tend to assign high similarity values too generously. For instance, if the correct solution is a 1-1 matching we expect each concept to be very similar (i.e., have high similarity value) to one concept at most, and very different (i.e., have low similarity value) to all others. Moreover, we want the similarity assignments to be stable in respect to the threshold value, so that changing the threshold slightly should not affect the final alignment considerably.

Therefore, given a matcher M and a concept c, we can define the *local confidence* of M with respect to c, $LC_M(c)$, as follows:

– let T be the set of all target concepts;

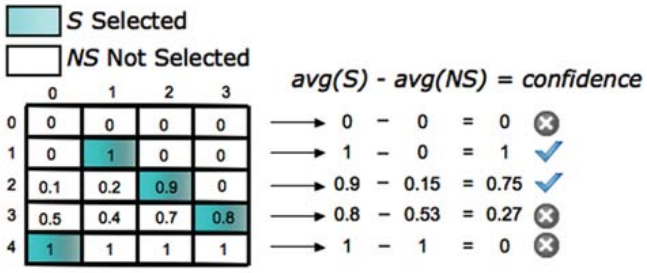


Fig. 6. Local confidence quality measure example.

- let $m_M(c) \subseteq T$ be the set of concepts $c' \in T$ that have been mapped to c by M;
- let $sim_M(c, c')$ be the similarity value between c and c' assigned by M;
- then $LC_M(c)$ is defined as the difference between the average of selected mappings' similarities for c and the average of the remaining correspondences' similarities:

$$LC_M(c) = \frac{\sum_{c' \in m_M(c)} sim_M(c, c')}{\mid m_M(c) \mid} - \frac{\sum_{c' \in (T - m_M(c))} sim_M(c, c')}{\mid T - m_M(c) \mid}.$$

With the reasonable assumption that M maps the most similar concepts, then $1 \geq \frac{\sum_{c' \in m_M(c)} sim_M(c, c')}{|m_M(c)|} \geq \frac{\sum_{c' \in (T - m_M(c))} sim_M(c, c')}{|T - m_M(c)|} \geq 0$, therefore $LC_M(c) \in [0, 1]$

A simple application of this quality measure is shown in Figure 6.

6 Experimental Results

In this section, we first report on the efficiency tests of the mappings selection algorithm. Then we compare the first layer matchers proposed in this paper (i.e., PSM and VMM) with the matching methods we used in the OAEI 2007 competition (i.e., BSM followed by DSI) [6]. Finally, we report on the results of the evaluation of the LWC matcher. We benefited from the capabilities of the AgreementMaker itself to perform the evaluations.

Mappings selection The most relevant module in this component is the 1-1 matching algorithm, which is also used to solve the *n-m* matching case. We compare the algorithm that we have adapted and implemented, the *Maximum Weight Bipartite Matching, MWBM,* with the Hungarian method [12] used by other matching systems.[2] In the first experiment, we ran both algorithms on random similarity matrices of different sizes (i.e., from a 500×500 matrix to a 5000×5000 matrix) with a threshold value of 0.5. As shown in Figure 7, the Hungarian method is much slower and uses a larger amount of memory.

In the second experiment, we investigated the effects of the threshold value on the performance of the algorithms. This time, we ran both methods on the same

[2] The implementation is available at konstantinosnedas.com/dev/soft/munkres.htm.

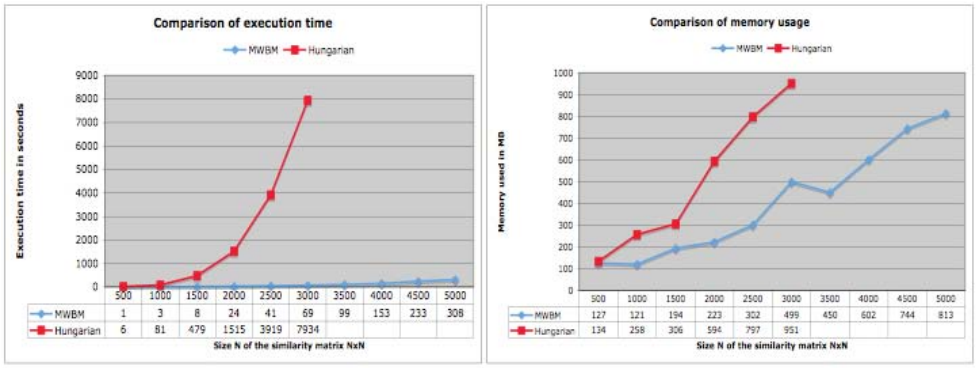


Fig. 7. Performance comparison between the Maximum Weight Bipartite Matching and the Hungarian method on different input sizes with a memory limitation of 1GB.

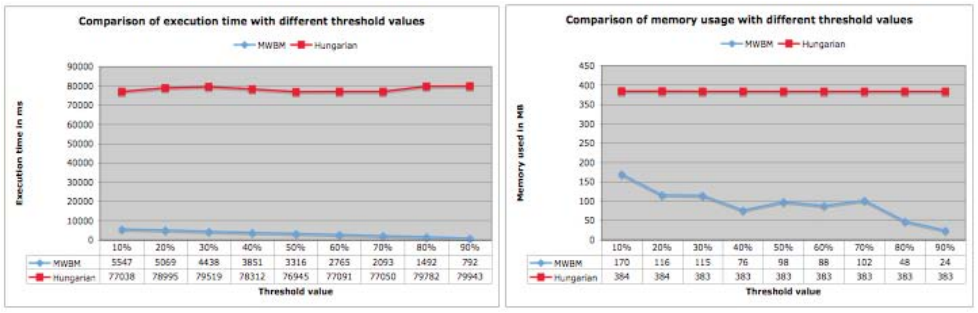


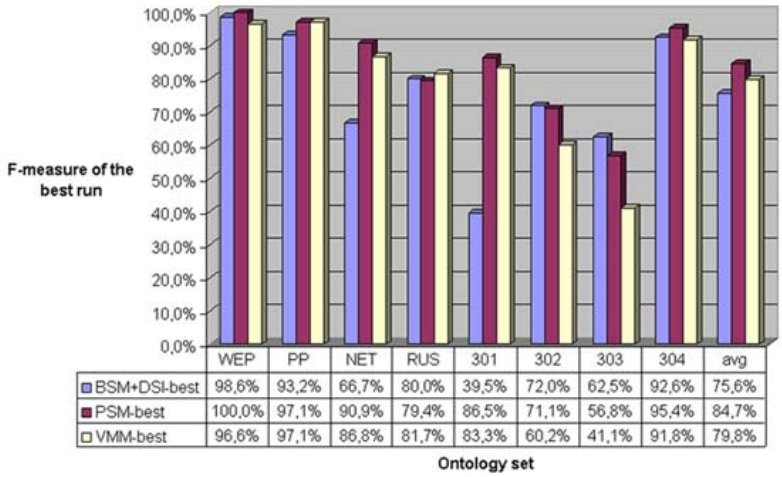
Fig. 8. Performance comparison between the Maximum Weight Bipartite Matching and the Hungarian method on different threshold values.

1000×1000 matrix using different threshold values (varying from 10% to 90%). As shown in Figure 8, the Hungarian method is not affected by the differences in the threshold values while the performance of MWBM improves when the threshold increases. That is, combinatorial matching methods, such as the Hungarian method, process the whole similarity matrix including those values that do not satisfy the threshold constraint. Instead, our algorithm transforms the similarity matrix into a weighted bipartite graph whose size is directly affected by the threshold value. Indeed, those correspondences that do not satisfy the threshold constraint are not translated into edges of the bipartite graph.

First layer matchers We ran the first two experiments on the alignment of eight pairs of ontologies. In particular, each set contains a source ontology, a target ontology, and the reference matching (expected matching) between them. The following ontology pairs were provided by I^3CON 2004:[3]

- **weapons set (WEP)** contains two classifications of various weapon types;
- **people and pets set (PP)**, contains two ontologies describing people and pets;
- **networks set (NET)** contains two classifications of computer networks;
- **Russia set (RUS)** contains general information about Russia.

[3] www.atl.external.lmco.com/projects/ontology/i3con.html



	WEP	PP	NET	RUS	301	302	303	304	avg
BSM+DSI-best	98,6%	93,2%	66,7%	80,0%	39,5%	72,0%	62,5%	92,6%	75,6%
PSM-best	100,0%	97,1%	90,9%	79,4%	86,5%	71,1%	56,8%	95,4%	84,7%
VMM-best	96,6%	97,1%	86,8%	81,7%	83,3%	60,2%	41,1%	91,8%	79,8%

Fig. 9. Comparison of first layer matchers (best run is considered).

The other four sets of ontologies are part of the OAEI benchmark.[4] The domain of these ontologies is bibliographic references. We consider those test cases in which the reference ontology *#101* has to be aligned with the following real-world ontologies:

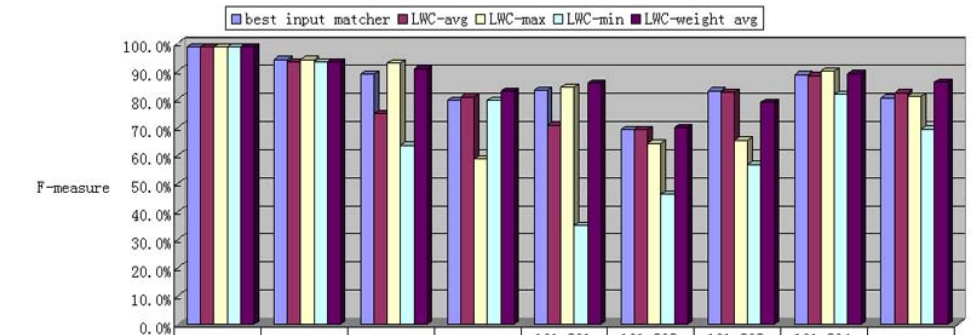
- **#301** is the BibTex bibliographic ontology from MIT;
- **#302** is the BibTex bibliographic ontology from UMBC;
- **#303** is the Karlsruhe bibliographic ontology used in the OntoWeb portal;
- **#304** is the INRIA bibliographic ontology.

In the first experiment, we ran the first layer matchers on all sets of ontologies using multiple threshold values for all of them. In Figure 9, for each method and for each ontology set, we report on the best F-measure of all runs. PSM is usually more effective than the others except for test cases #302 and #303, where it is slightly worse. However, the overall F-measure is definitely the highest. We further investigated the result of this experiment and noticed that BSM followed by DSI is quite accurate (high precision) but is able to find mappings only when the concepts are quite similar (otherwise displays low recall on dissimilar ontologies). Instead, PSM usually finds more mappings, even though some of them may be wrong occasionally. That is why it is less effective than BSM on the #303 set which contains mainly trivial mappings. VMM is sometimes better than the combination BSM+DSI, but it is usually worse than PSM. The problem is that these ontologies do not provide enough information to allow for this matcher to be very effective; however, it finds some non-trivial mappings not discovered by the other methods.

In summary, PSM is quite effective and stable, BSM is important for his high accuracy and VMM is able to find non-trivial mappings. Given the different qualities demonstrated by these matchers, we thought of combining them, thus motivating our next experiment.

LWC matcher We ran the first layer matchers (BSM, PSM, and VMM) and combined their results with the LWC matcher using four different linear operations: average of similarities, *LWC-avg*, maximum similarity, *LWC-max*, minimum similarity, *LWC-min*, and quality-based weighted average of similarities, *LWC-weight avg*.

[4] oaei.ontologymatching.org/2009/benchmarks/



	wep	pp	net	rus	101-301	101-302	101-303	101-304	avg
best input matcher	98.6%	94.2%	88.9%	79.7%	83.2%	69.2%	83.0%	88.8%	80.6%
LWC-avg	98.6%	93.2%	75.0%	80.9%	70.8%	69.2%	82.5%	88.5%	82.3%
LWC-max	98.6%	94.2%	92.9%	58.9%	84.2%	64.3%	65.5%	90.1%	81.1%
LWC-min	98.6%	93.2%	63.6%	79.7%	35.1%	46.2%	56.8%	81.8%	69.4%
LWC-weight avg	98.6%	93.3%	90.9%	82.9%	85.7%	69.9%	78.9%	89.2%	86.2%

Fig. 10. Comparison of combination techniques.

In the quality-based strategy, we adopted the *local confidence* quality to measure the relevance of the mappings generated by each matcher. In particular, the LWC method computes a combined similarity matrix that is obtained as the weighted average of the similarity matrices produced by the three matchers (see Figure 4). In this experiment, the weights are assigned by evaluating each matcher with the *local confidence* quality measure. Being a *local similarity* level quality measure (see Table 1), it defines a different value for each row of a similarity matrix, which is directly used to compute the weighted average for that row in the combination of the similarity matrices.

For each ontology set, we report in Figure 10 the best performance of the matchers to be combined, henceforth called *input matchers*, and the performance of the different versions of the LWC matcher. In most cases, almost all the mappings found by a single matcher are included in the set generated by another one, therefore any combination of these matchers cannot provide a significant improvement. A combined result equivalent to the best matcher in the input is already a good result. However, in all test cases, at least one of the combined results is equivalent to or better than the best result of the input matchers.

Considering the complexity of combining multiple matchings, which is still an open research problem, the most important result of this experiment is that the weighted average based on the local confidence quality is the most effective technique. Moreover, we note that the weights are chosen *automatically*.

7 Conclusions

In this paper, we make a contribution to the automatic evaluation of matchings by defining a quality measure that does not take into account the prior knowledge of a reference matching. We use this quality measure to define the weighting scheme of a *fully automatic* combination method. We also propose an efficient solution for the mappings selection task, whose performance is also positively affected by the threshold value. We plan to provide an API to make this functionality available to other matching systems.

In the future, we will take advantage of our extensible architecture and add new matching methods, for example, to our instance based methods. We plan to study new quality measures to enhance the current evaluation capabilities and our quality-based combination technique. Another direction for future research includes using partial reference matchings to perform the alignment of full ontologies.

References

1. I. F. Cruz, F. Palandri Antonelli, and C. Stroe. AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. *PVLDB*, 2(2):1586–1589, 2009.
2. I. F. Cruz, F. Palandri Antonelli, and C. Stroe. Integrated Ontology Matching and Evaluation. In *International Semantic Web Conference (Posters & Demos)*, 2009. To appear.
3. I. F. Cruz, A. Rajendran, W. Sunna, and N. Wiegand. Handling Semantic Heterogeneities using Declarative Agreements. In *ACM Symposium on Advances in Geographic Information Systems (ACM GIS)*, pages 168–174, 2002.
4. I. F. Cruz and W. Sunna. Structural Alignment Methods with Applications to Geospatial Ontologies. *Transactions in GIS, Special Issue on Semantic Similarity Measurement and Geospatial Applications*, 12(6):683–711, December 2008.
5. J. Euzenat. Semantic Precision and Recall for Ontology Alignment Evaluation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 348–353, 2007.
6. J. Euzenat, A. Isaac, C. Meilicke, P. Shvaiko, H. Stuckenschmidt, O. Šváb, V. Svátek, W. R. van Hage, and M. Yatskevich. Results of the Ontology Evaluation Initiative 2007. In *ISWC International Workshop on Ontology Matching (OM)*, volume 304, pages 96–132. CEUR-WS, 2007.
7. J. Euzenat, M. Mochol, P. Shvaiko, H. Stuckenschmidt, V. Svátek, W. R. van Hage, and M. Yatskevich. Results of the Ontology Alignment Evaluation Initiative. In *ISWC International Workshop on Ontology Matching (OM)*, volume 225, pages 73–95. CEUR-WS, 2006.
8. J. Euzenat and P. Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.
9. N. Jian, W. Hu, G. Cheng, and Y. Qu. Falcon-AO: Aligning Ontologies with Falcon. In *K-CAP 2005 Workshop on Integrating Ontologies*, volume 156, pages 85–91. CEUR-WS, 2005.
10. C. Joslyn, A. Donaldson, and P. Paulson. Evaluating the Structural Quality of Semantic Hierarchy Alignments. In *International Semantic Web Conference (Posters & Demos)*, volume 401. CEUR-WS, 2008.
11. R. M. Karp. An Algorithm to Solve the $m \times n$ Assignment Problem in Expected Time $O(mn \log n)$. *Networks*, 10(2):143–152, 1980.
12. H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.
13. P. Lambrix, H. Tan, and Q. Liu. SAMBO and SAMBOdtf Results for the Ontology Alignment Evaluation Initiative 2008. In *ISWC International Workshop on Ontology Matching (OM)*, volume 431. CEUR-WS, 2008.
14. C. Meilicke and H. Stuckenschmidt. Incoherence as a Basis for Measuring the Quality of Ontology Mappings. In *ISWC International Workshop on Ontology Matching (OM)*, volume 431. CEUR-WS, 2008.
15. J. Tang, J. Li, B. Liang, X. Huang, Y. Li, and K. Wang. Using Bayesian Decision for Ontology Mapping. *Journal of Web Semantics*, 4(4):243–262, 2006.
16. O. Udrea, L. Getoor, and R. J. Miller. Leveraging Data and Structure in Ontology Integration. In *ACM SIGMOD International Conference on Management of Data*, pages 449–460, 2007.

Measuring the Structural Preservation of Semantic Hierarchy Alignments

Cliff A Joslyn, Patrick Paulson, and Amanda White

Pacific Northwest National Laboratory, Richland, WA, USA
{cjoslyn,patrick.paulson,amanda.white}@pnl.gov

Abstract. We present a method to measure the amount of structural distortion carried by an alignment between two taxonomic cores of ontologies represented as semantic hierarchies. We present our formalism based in metric order theory. We then illustrate the results of such an analysis on the Anatomy track of the 2008 Ontology Alignment Evaluation Initiative (OAEI).

Key words: Ontology alignment; lattice theory; order theory.

1 Introduction

Since top-down, monolithic development of unitary ontologies is at best difficult, and at worst undesirable, ontology alignment is increasingly seen as a critical Semantic Web technology [4, 17]. Although many semantic relations can be present in ontologies, they tend to be dominated by their taxonomic cores; that is, subsumptive inheritance (`is-a`) and/or meronomic compositional (`part-of`) class hierarchies. Thus techniques which address the specific nature of these structures as *semantic hierarchies* are critical for ontology management tasks.

An alignment is modeled as a mapping (single- or multi-valued) between two semantic hierarchies, taking concepts from one into another. Depending on the relative size, structure, and domains of the two hierarchies, their quality, and the size and quality of the alignment, different properties of the alignment might hold. It might be that that mapping is partial in one direction or the other; it may be concentrated in one portion or another of each hierarchy; may takes nodes which are "close together" in one hierarchy into nodes which are "far apart" in the other; and may take nodes in a particular structural relationship (e.g. parent-child or sibling) into the same or a different such structural relationship. Knowledge of such properties is valuable for the ontology designer and aligner, an important adjunct to visual inspection of large ontologies and alignments.

One straightforward example of this reasoning is to say that if the two semantic hierarchies were intended to model the same domain, then an alignment mapping should be structure-preserving, taking pairs of nodes which are close together in one structure into pairs which are also close together in the other, and similarly for pairs of nodes which are far apart. To the extent that this is not the case, this could indicate a problem with either one ontology, the other, the alignment mapping, or some combination of these structures. Even when semantic or pragmatic criteria dictate that it is appropriate for a mapping to violate structural preservation, it is still valuable to be able to *measure* and *quantify* the amount of structural preservation or distortion which an alignment introduces.

This is true both after the alignment has been produced, and also *while* the alignment is being produced, for example in an interactive environment such as the Protege tool PROMPT [17].

We describe an algorithmic approach to the measurement of the extent to which an ontology alignment preserves the structural properties of the two ontologies. We use **order theory** (the formal theory of hierarchy represented by ordered sets and lattices [3]) to model taxonomies as semantic hierarchies on sets of nodes P, where nodes $a \in P$ are ontology concepts related by transitive edges such as subsumption ("`is-a`") or composition ("`part-of`"). These in turn are represented as finite, bounded, partially ordered sets (posets) $\mathcal{P} = \langle P, \leq \rangle$, where the relation $\leq$ is one (or a union) of these transitive link types. Such ordered structures are not, in general, trees, nor even lattices, but can be rich in multiple inheritance and lack unique least common subsumers between nodes.

We demonstrate our approach by analyzing the alignments of the Anatomy track of the 2008 Ontology Alignment Evaluation Initiative (OAEI) campaign (http://oaei.ontologymatching.org/2008/anatomy). We compare the precision and recall results of the OAEI against our discrepancy measures, as well as analyze the highest discrepancy nodes and alignment links.

Prior work in both ontology alignment in general, and graph matching in knowledge systems (e.g. [7]), is voluminous, and order theory is used in many areas of computer science outside of knowledge systems. But there is relatively little in the ontology literature about measuring structural relations in ontologies, and we've been able to find nothing in the specific use of a lattice theoretical approach to hierarchy mapping and measurement. Kalfoglou and Schorlemmer [12] have an approach to order morphisms similar to ours; and some researchers [5, 17] take a structure mapping approach, but do so as a graph theory problem, not using hierarchy theory. Although He and Xiaoyong [8] recognize the need to work in order theory, they don't actually do so.

The algebraic relations among class extents and intents used by a number of researchers (e.g. [14, 15]) do point to metric properties similar to ours. But while these have implications for an order-theoretical approach, they are not themselves explicitly order-theoretical. The closest correlate to our order metric approach is in the use of "semantic similarity" measures [1]. Still, these are generally used *within* a particular lexical or bio-ontology, and have only been used to a small extent [19] as an adjunct to the alignment problem. Some of our work [10] marries structural similarities with our order metrics. We are actively working [18] to identify how our order metrics are actually foundational to semantic similarities, and generate them as a special case.

An early desription of this concept has been previously reported in a poster [11].

2 Order Theory for Semantic Hierarchy Alignment

We represent semantic hierarchies as bounded, partially ordered sets (posets) $\mathcal{P} = \langle P, \leq \rangle$ [3], where P is a finite set of ontology nodes, and $\leq \subseteq P^2$ is a reflexive, anti-symmetric, and transitive binary relation such as subsumption ("`is-a`")

or composition ("`part-of`"). In ontology analysis, semantic hierarchies are typically Directed Acyclic Graphs (DAGs) [9] which are top-bounded, have a moderate amount of multiple inheritance, and branch downward very strongly. Each such structure uniquely determines a poset $\mathcal{P}$ by taking its transitive closure and including a bottom bound $0 \in P$ such that $\forall a \in P, 0 \leq a$.

For two taxonomies $\mathcal{P} := \langle P, \leq \rangle, \mathcal{P}' := \langle P', \leq' \rangle$, an **alignment relation** $F \subseteq P \times P'$ is a collection of pairs $\boldsymbol{f} = \langle a, a' \rangle \in F$, indicating that the node $a \in P$ on the "left" side is mapped or aligned to the node $a' \in P'$ on the "right" side. F determines a domain and codomain

$$Q := \{a \in P, \exists a' \in P', \langle a, a' \rangle \in F\} \subseteq P, \quad Q' := \{a' \in P', \exists a \in P, \langle a, a' \rangle \in F\} \subseteq P',$$

We call the $\boldsymbol{f} \in F$ **links**, the $a \in Q$ the **left anchors** and the $a' \in Q'$ the **right anchors**. Let $m := |Q|, m' := |Q'|$, and $N := |F| \leq mm'$.

Fig. 1 shows a small alignment. We have left anchors $Q = \{B, E, G\}, m = 3$; right anchors $Q' = \{I, J, K\}, m' = 3$; and $N = 4$ with links $F = \{\boldsymbol{f}_1 = \langle B, J \rangle, \boldsymbol{f}_2 = \langle B, I \rangle, \boldsymbol{f}_3 = \langle E, I \rangle, \boldsymbol{f}_4 = \langle G, K \rangle\}$.

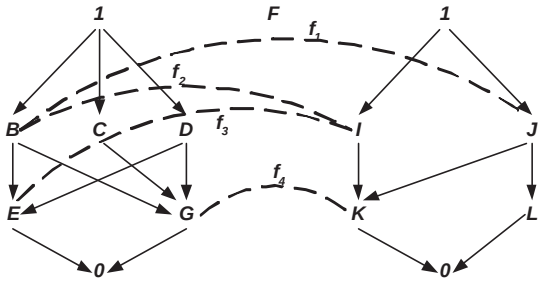


Fig. 1. An example of two semantic hierarchies and an alignment relation.

Let d be a metric on $\mathcal{P}$ and $\mathcal{P}'$. For links $\boldsymbol{f} = \langle a, a' \rangle, \boldsymbol{g} = \langle b, b' \rangle \in F$ to participate well in a good structural mapping between $\mathcal{P}$ and $\mathcal{P}'$, we want the metric relations between the $a, b \in Q$ to be the same as their corresponding $a', b' \in Q'$, so that $|\bar{d}(a, b) - \bar{d}'(a', b')|$ is small. In our example, F takes both B and E, which are somewhat distant in $\mathcal{P}$, to the single node I in $\mathcal{P}'$, so that there is no distance between them on the right. This is not preferred.

We now consider our metric d needed to compare the distances $d(a, b), d(a', b')$ between pairs of nodes $a, b \in P$ on one side of an alignment and their images $a', b' \in P$ on another. The knowledge systems literature has focused on **semantic similarities** [1] to perform a similar function, which are available when $\mathcal{P}$ is equipped with a probabilistic weighting function p: $P \to [0, 1]$, with $\sum_{a \in P} p(a) = 1$. p can be derived, for example, from the frequency with which terms appear in documents (for the case of the Wordnet [6] thesaurus), or which genes are annotated to bio-ontology nodes (in the case of the Gene Ontology [13]).

Our purpose is more general, since we may not have such a weighting function available, and semantic similarities are not required to be metrics satisfying the triangle inequality. In seeking out the proper mathematical grounding, we turn to **order metrics** [16, 18] which can use, but do not require, a quantitative

weighting, and always yield a metric. For details about order metrics built from isotone and antitone lower and upper semimodular functions on ordered sets, see [18]. In this work, we use the **upper and lower cardinality-based distances**

$$d_u(a,b) = |\uparrow a| + |\uparrow b| - 2\max_{c \in a \vee b} |\uparrow c|, \qquad d_l(a,b) = |\downarrow a| + |\downarrow b| - 2\max_{c \in a \wedge b} |\downarrow c|,$$

where for a node $a \in P$, its **upset** $\uparrow a := \{b \geq a\}$ and **downset** $\downarrow a := \{b \leq a\}$ are all its ancestors and successors respectively, so that $|\uparrow a|, |\downarrow a|$ are the number of ancestors and successors. The generalized join and meet are

$$a \vee b := \mathrm{Min}(\uparrow a \cap \uparrow b) \subseteq P, \qquad a \wedge b := \mathrm{Max}(\downarrow a \cap \downarrow b) \subseteq P,$$

where for a set of nodes $Q \subseteq P$ the **upper bounds** and **lower bounds** are $\mathrm{Min}(Q) := \{a \in Q : \nexists b \in Q, b < a\} \subseteq P$, $\mathrm{Max}(Q) := \{a \in Q : \nexists b \in Q, b > a\} \subseteq P$.

We need to normalize distance to the size of the structure, so that we are measuring the relative proportion of the overall structure two nodes are apart, or in other words, what proportion of their potential maximum distance. These **normalized upper and lower distances** are

$$\bar{d}_u(a,b) := \frac{d_u(a,b)}{|P|-1} \in [0,1], \quad \bar{d}_l(a,b) := \frac{d_l(a,b)}{|P|-1} \in [0,1].$$

Considering the difference between upper and lower distance, it may at first appear to be more natural to use upper distance, since we're then "looking upwards" towards the top bound $1 \in P$ which almost always exists in the given structure. Moreover, it is sometimes the case that the upper distance $d_u(a,b)$ is the same as the minimum (undirected) path length between a and b (a favorite graph metric), but this is only required to be true when $\mathcal{P}$ is an upper-bounded tree: in general, path length and these metrics are unrelated.

When $\mathcal{P}$ is top-bounded and strongly down-branching (as in our cases), then it is preferable to use lower distance (this is possible because we always provide a lower bound $0 \in P$). One reason for this is that since semantic hierarchies are much more strongly down-branching than up-branching, up-sets are typically very small and narrow, frequently single chains; where down-sets are large, branching structures. Additionally, this allows siblings deep in the hierarchy to be closer together than siblings high in the hierarchy (this will be demonstrated below). This is considered valuable, for example, where e.g. "mammal" and "reptile" are considered farther apart than "horse" and "goat".

In Fig. 1, to calculate the lower distance $d_l(B,C)$, we have $|\downarrow B| = 4, |\downarrow C| = 3, B \wedge C = \{G, 0\}$, $\max_{c \in B \wedge C} |\downarrow c| = \max(1,2)$, so that $d_l(B,C) = 4 + 3 - 2 \times 2 = 3$. Finally, we have $|P| = 7$, so that $\bar{d}_l(B,C) = 1/2$. Table 1 shows distances $d_l(a,b)$ on the left in $\mathcal{P}$, and Table 2 shows distances $d_l(a',b')$ on the right in $\mathcal{P}'$. $|P| = 6$ and $|P'| = 5$, yielding Tables 3 and 4 showing the relative distances. Note that siblings high in the structure are farther apart than those lower, for example $\bar{d}_l(B,C) = 0.50, \bar{d}_l(E,G) = 0.33$, and $\bar{d}_l(I,J) = 0.60, \bar{d}_l(K,L) = 0.40$. Contrast this with the similar relative *upper* distances, shown in Table 5, where siblings lower in the structure are further apart.

$d_l(a,b)$	1	B	C	D	E	G	0
1	0	3	4	3	5	5	6
B	3	0	3	4	2	2	3
C	4	3	0	3	3	1	2
D	3	4	3	0	2	2	3
E	5	2	3	2	0	2	1
G	5	2	1	2	2	0	1
0	6	3	2	3	1	1	0

Table 1. Left lower distances $d_l(a,b)$

$d_l(a',b')$	1	I	J	K	L	0
1	0	3	2	4	4	5
I	3	0	3	1	3	2
J	2	3	0	2	2	3
K	4	1	2	0	2	1
L	4	3	2	2	0	1
0	5	2	3	1	1	0

Table 2. Right lower distances $d_l(a',b')$.

$\bar{d}_l(a,b)$	1	B	C	D	E	G	0
1	0.00	0.50	0.67	0.50	0.83	0.83	1.00
B	0.50	0.00	0.50	0.67	0.33	0.33	0.50
C	0.67	0.50	0.00	0.50	0.50	0.17	0.33
D	0.50	0.67	0.50	0.00	0.33	0.33	0.50
E	0.83	0.33	0.50	0.33	0.00	0.33	0.17
G	0.83	0.33	0.17	0.33	0.33	0.00	0.17
0	1.00	0.50	0.33	0.50	0.17	0.17	0.00

Table 3. Left lower relative distances $\bar{d}_l(a,b)$

$\bar{d}_l(a',b')$	1	I	J	K	L	0
1	0.00	0.60	0.40	0.80	0.80	1.00
I	0.60	0.00	0.60	0.20	0.60	0.40
J	0.40	0.60	0.00	0.40	0.40	0.60
K	0.80	0.20	0.40	0.00	0.40	0.20
L	0.80	0.60	0.40	0.40	0.00	0.20
0	1.00	0.40	0.60	0.20	0.20	0.00

Table 4. Right lower relative distances $\bar{d}_l(a',b')$

Let d be a metric used in both $\mathcal{P}, \mathcal{P}'$, in our case, the lower distance d_l. Then the **link discrepancy** is given by $\delta(\boldsymbol{f}, \boldsymbol{g}) := |\bar{d}(a,b) - \bar{d}(a',b')|$, and the **distance discrepancy induced by F between $\mathcal{P}$ and $\mathcal{P}'$** given d is

$$D(F) := \frac{\sum_{\boldsymbol{f},\boldsymbol{g} \in F} \delta(\boldsymbol{f}, \boldsymbol{g})}{\binom{N}{2}}.$$

$D \in [0,1]$, with $D = 0$ iff F is completely distance preserving, and $D = 1$ if F is maximally distance distorting, e.g. mapping diameters to equality, and neighbors and children to diameters. Table 7 shows the discrepancies δ comparing links against each other, yielding total distance discrepancy $D(F) = 0.26$.

$\bar{d}_u(a,b)$	1	B	C	D	E	G	0
1	0.00	0.17	0.17	0.17	0.50	0.67	1.00
B	0.17	0.00	0.33	0.33	0.33	0.50	0.83
C	0.17	0.33	0.00	0.33	0.67	0.50	0.83
D	0.17	0.33	0.33	0.00	0.33	0.50	0.83
E	0.50	0.33	0.67	0.33	0.00	0.83	0.50
G	0.67	0.50	0.50	0.50	0.83	0.00	0.33
0	1.00	0.83	0.83	0.83	0.50	0.33	0.00

Table 5. Left upper relative distances $\bar{d}_u(a,b)$

$\bar{l}(a,b)$	1	B	C	D	E	G	0
1	0.00	0.33	0.33	0.33	0.67	0.67	1.00
B	0.33	0.00	0.67	0.67	0.33	0.33	0.67
C	0.33	0.67	0.00	0.67	1.00	0.33	0.67
D	0.33	0.67	0.67	0.00	0.33	0.33	0.67
E	0.67	0.33	1.00	0.33	0.00	0.67	0.33
G	0.67	0.33	0.33	0.33	0.67	0.00	0.33
0	1.00	0.67	0.67	0.67	0.33	0.33	0.00

Table 6. Normalized minimum undirected path length.

We wish to understand the contribution which particular links and anchors make to the overall discrepancy. So we aggregate discrepancies over links $\boldsymbol{f}, \boldsymbol{g} \in F$, normalized by the number of links; and over left and right anchors $a \in Q, a' \in Q'$, normalized by the number of left and right anchors respectively (results for our example are shown in Tables 8 and 9):

	$\boldsymbol{f}_1 = \langle B, J \rangle$	$\boldsymbol{f}_2 = \langle B, I \rangle$	$\boldsymbol{f}_3 = \langle E, I \rangle$	$\boldsymbol{f}_4 = \langle G, K \rangle$
$\boldsymbol{f}_1 = \langle B, J \rangle$	0.00	0.60	0.27	0.07
$\boldsymbol{f}_2 = \langle B, I \rangle$	0.60	0.00	0.33	0.13
$\boldsymbol{f}_3 = \langle E, I \rangle$	0.27	0.33	0.00	0.13
$\boldsymbol{f}_4 = \langle G, K \rangle$	0.07	0.13	0.13	0.00

Table 7. Distance discrepancy $\delta(\boldsymbol{f}_i, \boldsymbol{f}_j)$.

$$D(\boldsymbol{f}) := \frac{\sum_{\boldsymbol{g} \in F} \delta(\boldsymbol{f}, \boldsymbol{g})}{N-1},\ D(a) := \frac{\sum_{\langle a, a' \rangle \in F} D(\langle a, a' \rangle)}{m},\ D(a') := \frac{\sum_{\langle a, a' \rangle \in F} D(\langle a, a' \rangle)}{m'}.$$

Because we use lower distance, links high in the structure are further apart, for example $\delta(\langle B, I \rangle, \langle B, J \rangle) = 0.60$, since the identical pair $\langle B, B \rangle$ which are zero apart are taken to the nodes $\langle I, J \rangle$ high in the structure; while $\delta(\boldsymbol{f}_1, \boldsymbol{f}_4) = 0.07$, since $\langle B, G \rangle$ are almost as close on the left as $\langle J, K \rangle$ on the right. The link $\boldsymbol{f}_2 = \langle B, I \rangle$ is the greatest contributor to distance discrepancies, as are its anchors $B \in P, I \in P'$. This result is slightly counterintuitive, but instructive. Considering *link* comparisons in Fig. 1: comparing $\boldsymbol{f}_1$ to $\boldsymbol{f}_3$, for example, the *differences* in the distances between their left and right anchors is smaller than the similar difference comparing the left and right anchors $\boldsymbol{f}_2$ and $\boldsymbol{f}_3$.

3 Analysis of the 2008 OAEI Anatomy Track

We now describe the application of this alignment evaluation technology against the Anatomy track of the 2008 OAEI campaign [2]. In the OAEI, one or more "gold standard" reference alignments are developed (in part, by hand) between pairs of ontologies. The community is challenged to submit alignments, and their quality is measured by calculating the precision, recall, and F-score of the matches between nodes made by the submitted alignments against those matches made by the reference alignment. We calculated distance discrepancies for the alignments in the challenge track, including the references. We compared the discrepancy scores of the submitted alignments to each other, and to the references, and correlated the precision and recall results of the submitted alignments against their discrepancies.

	$D(\boldsymbol{f}_i)$
$\boldsymbol{f}_1 = \langle B, J \rangle$	0.31
$\boldsymbol{f}_2 = \langle B, I \rangle$	0.35
$\boldsymbol{f}_3 = \langle E, I \rangle$	0.24
$\boldsymbol{f}_4 = \langle G, K \rangle$	0.11

Table 8. Aggregate distance discrepancy by link $D(\boldsymbol{f})$.

	I	J	K	$D(a)$
B	0.35	0.31		0.22
E	0.24			0.08
G			0.11	0.04
$D(a')$	0.20	0.10	0.04	

Table 9. Aggregate distance discrepancy by anchor $D(a), D(a')$.

3.1 The Anatomy Track Ontologies and Alignments

The OAEI-2008 Anatomy track was selected due to its sufficient size, moderate amount of complexity and multiple inheritance, and publicly available partial reference alignment. It included the 2744 classes of the Adult Mouse Anatomy

(MA, http://www.informatics.jax.org/searches/AMA_form.shtml) and the portion of the 3304 classes from the NCI Thesaurus (NCIT)[1] describing human anatomy.

The full reference alignment was provided by the Mouse Genome Informatics group at the Jackson Laboratory. The partial reference alignment had 988 links, derived from 934 purely lexical matches and 54 additional links from the full reference (http://oaei.ontologymatching.org/2008/results/anatomy). There were multiple tasks in the anatomy track, and we focused on Task 1, which was to maximize the F-score of an alignment. We also focused on the nine submitted alignments with code names shown in Table 10.

A statistical analysis of MA and NCIT shows structures which are somewhat similar in size (2744 vs. 3304 nodes, respectively), "edge density" (1.04 vs. 1.14 links/edge), and "leaf density" (82.3% vs. 79.6% of the nodes being leaves). But MA is dramatically shorter, with a height (maximum chain length from the top 1 to bottom 0) of 8 compared to 14. NCIT is more complex, with dramatically more multiple inheritance (4.0% vs. 13.2% of nodes with more than one parent).

3.2 Discrepancy Measurement Results

Table 10 lists basic statistics for all alignments, and also shows the number of anchors and links, the discrepancies, and (for the submitted alignments) the precision P, recall R, and F-score $2PR/(P+R)$. Fig. 2 shows distance discrepancies with the number of anchors and links on the right Y axis; Fig. 3 shows them with P, R and F-score on the right Y axis.

Alignment	m	m'	N	$D(F)$	P	R	F-score
reference_partial	986	984	988	0.08%			
reference_full	1501	1504	1523	0.11%			
aflood	1186	1186	1186	0.11%	87.4%	68.2%	76.6%
AROMA	1062	1062	1062	0.36%	80.3%	56.0%	66.0%
ASMOV	1261	1261	1261	0.09%	78.7%	65.2%	71.3%
DSSim	1170	1086	1545	2.02%	61.6%	62.4%	62.0%
Lily	1324	1324	1324	0.13%	79.6%	69.3%	74.1%
RiMOM	1205	1205	1205	0.10%	92.9%	73.5%	82.1%
SAMBO	1465	1465	1465	0.09%	86.9%	83.6%	85.2%
SAMBOdtf	1527	1527	1527	0.10%	83.1%	83.3%	83.2%
TaxoMap	2533	1279	2533	1.40%	46.0%	76.4%	57.4%

Table 10. Discrepancy results for anatomy track alignments. $D(F)$ = lower distance discrepancy; N = # links; P = precision; R = recall; F = F-score.

Generally, discrepancies are low, especially for the two reference alignments, except for AROMA, DSSIM, and Taxomap. These are also the worst performers, and DSSIM and Taxomap have the biggest difference between number of anchors and links. Fig. 4 shows distance discrepancy $D(F)$ against F-score. Significant discrepancy is an indication of poor F-score, and conversely high F-score requires effectively no discrepancy: Pearson correlation between D and F-score $-.780$.

Table 11 shows the top nine links by aggregate discrepancy for the partial and full reference alignments, and the two worst-scoring alignment by both F-score and discrepancy. As illustrated in Table 9, aggregation of discrepancy by anchor is most valuable when the alignment F is not very one-to-one. This is the

[1] http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview/vocabulary

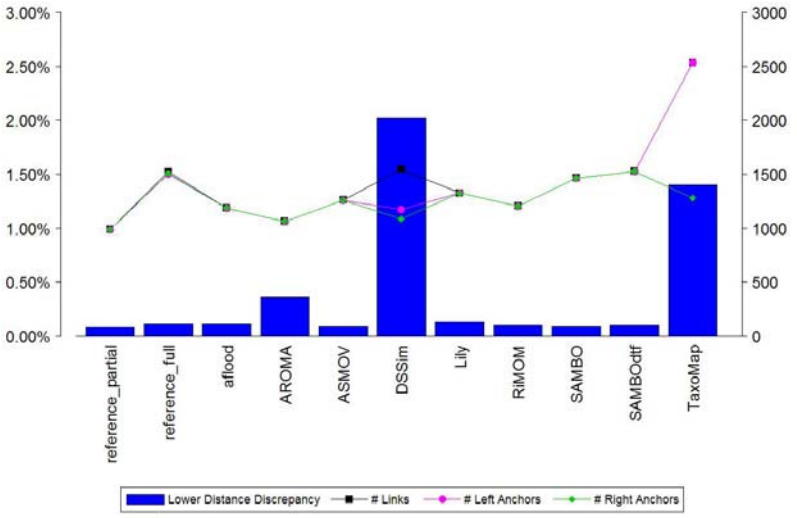


Fig. 2. Discrepancy against number of anchors and links.

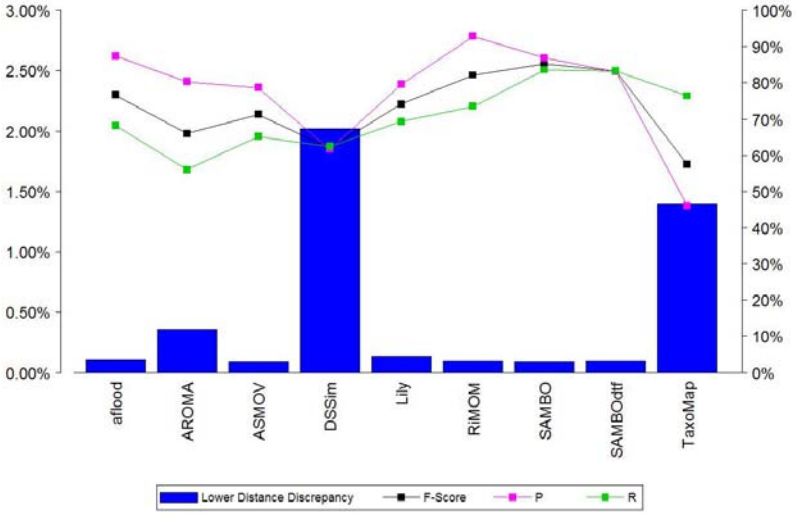


Fig. 3. Discrepancy against precision, recall, and F-score.

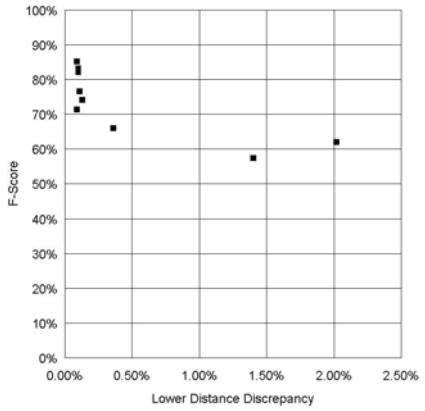


Fig. 4. Anatomy track alignments, distance discrepancy against F-score.

case with the two reference alignments, so Table 13 shows the top nine aggregate discrepancies by left- and right-anchors for DSSIM and Taxomap.

Partial Reference			Full Reference		
D	**MA**	**NCIT**	D	**MA**	**NCIT**
3.05%	organ system	Organ_System	17.30%	blood vessel	Blood_Vessel
2.81%	blood vessel	Blood_Vessel	6.78%	venous blood vessel	Venous_System
2.73%	vein	Vein	5.32%	skeletal muscle	Skeletal_Muscle_Tissue
1.53%	connective tissue	Connective_Tissue	3.07%	organ system	Organ_System
1.38%	bone	Bone	2.74%	vein	Vein
1.00%	artery	Artery	1.86%	limb bone	Bone_of_the_Extremity
0.97%	foot bone	Foot_Bone	1.68%	vertebra	Vertebra
0.71%	lymphoid tissue	Lymphoid_Tissue	1.57%	head/neck muscle	Head_and_Neck_Muscle
0.68%	ligament	Ligament	1.55%	connective tissue	Connective_Tissue
0.67%	muscle	Muscle	1.39%	bone	Bone

DSSIM			Taxomap		
D	**MA**	**NCIT**	D	**MA**	**NCIT**
62.82%	joint	Body_Part	11.72%	tail blood vessel	Blood_Vessel
15.66%	cardiovascular system	Cardiovascular_System_Part	11.72%	foot blood vessel	Blood_Vessel
13.02%	capillary	Blood_Vessel	11.72%	neck blood vessel	Blood_Vessel
11.04%	bone	Loose_Connective_Tissue	11.72%	head blood vessel	Blood_Vessel
9.84%	perineal artery	Perineal_Artery	11.72%	lung blood vessel	Blood_Vessel
9.84%	ethmoidal artery	Artery	11.72%	upper leg blood vessel	Blood_Vessel
8.87%	brachial artery	Brachial_Artery_Branch	11.72%	lower leg blood vessel	Blood_Vessel
8.84%	celiac artery	Artery	11.72%	pelvis blood vessel	Blood_Vessel
8.82%	radial artery	Artery	11.72%	abdomen blood vessel	Blood_Vessel

Table 11. Top nine aggregate link distance discrepancies $D(\boldsymbol{f})$ for four alignments.

Fig. 5 shows a selection of nodes from MA and NCIT, and anchors and links from both the partial and full reference alignments. Numbers below terminal nodes indicate the total number of nodes below them. The top three link discrepancies are shown in Table 12, with labels referring to particular links in Fig. 5. We can see that the biggest discrepancies are between links which take nodes high in MA to nodes low in NCIT. But in fact, our method does not count vertical ranks, but rather the order metrics focus on the numbers of common nodes below the corresponding pairs of anchors.

		f			*g*	
δ		MA	NCIT		MA	NCIT
19.8%	F*	blood vessel	Venous System	F3	skeletal muscle	Skeletal_Muscle_Tissue
17.6%	F*	blood vessel	Venous System	F1=P4	organ system	Organ_System
16.3%	F*	blood vessel	Venous System	F+	limb bone	Bone_of_the_Extremity

Table 12. Highest discrepancies between link pairs shown in Fig. 5, full reference alignment.

Comparing alignments now, while both reference alignments had low discrepancy, the full alignment was generally more discrepant, perhaps through the addition of non-lexical matching links like ⟨venous blood vessel, Venous_System⟩. In DSSIM, clearly the link ⟨joint, Body_Part⟩ is most discrepant. This is because while both Joint and Body_Part are relatively near the tops of MA and NCIT respectively, Joint covers only 21 nodes, while Body_Part covers 2137. This forces that link to be far from all the others, and reveals directly a dramatic difference in structure between the two ontologies. This is then reflected in a very high anchor aggregate score $D(\text{Body\_Part}) = 5.44$. Finally, for Taxomap, we see many links to the NCIT node Blood_Vessel, yielding another high anchor discrepancy of $D(\text{Blood\_Vessel}) = 9.48$. In both cases, the discrepancy measures can point directly to anomolous mappings of high significance.

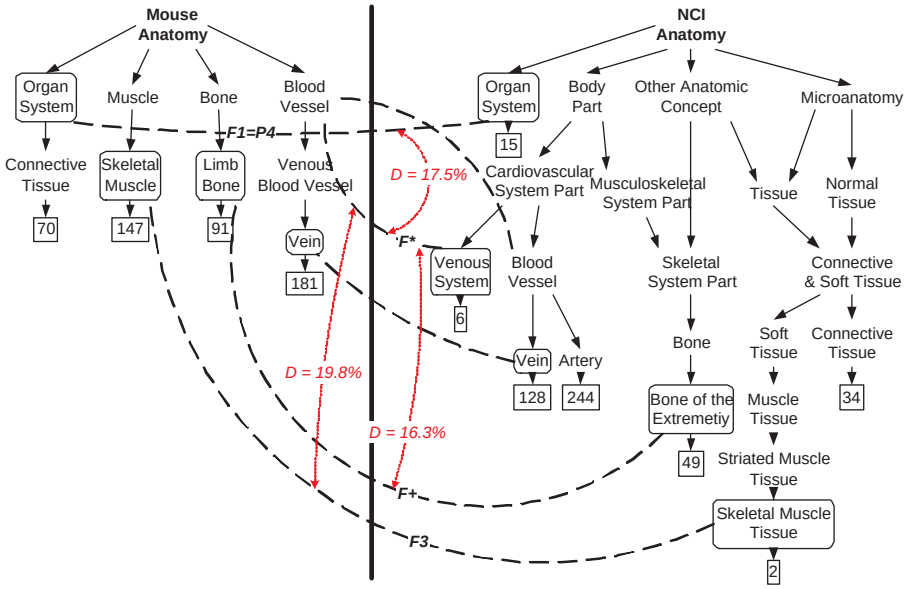


Fig. 5. Selection of nodes from MA and NCIT, and anchors and links from both the partial and full reference alignments. Table 12 provide details on comparisions of high discrepancy links.

4 Conclusions and Further Work

The results presented here are the first serious attempt to apply this technology to alignment analysis, and are partial and preliminary. Results here may be dependent on the particular properties of the Anatomy track. While a further analysis relating alignment quality to discrepancy awaits, it is suggestive that a discrepancy analysis can reveal to the aligner and ontology designer aspects of their structures not clear from visual inspection. Nor is a robust order theoretical technology limited to discrepancy measures: we can see above that other considerations such as the degree to which alignments are many-to-many, vertical rank structure, degree of multiple inheritance, and a range of other topics in interaction with discrepancies awaits much more serious consideration.

DSSIM				**Taxomap**			
MA		**NCIT**		**MA**		**NCIT**	
D	a	D	a'	D	a	D	a'
0.830	joint	5.440	Body Part	0.117	tail blood vessel	9.483	Blood Vessel
0.207	cardiovascular system	2.836	Artery	0.117	foot blood vessel	4.093	Muscle
0.172	capillary	1.893	Vein	0.117	neck blood vessel	3.418	Vein
0.165	skeletal muscle	0.985	Bone	0.117	head blood vessel	3.356	Artery
0.146	bone	0.614	Blood Vessel	0.117	lung blood vessel	1.413	Bone
0.130	perineal artery	0.273	Loose Connective Tissue	0.117	upper leg blood vessel	1.215	Connective Tissue
0.130	ethmoidal artery	0.257	Skeletal Muscle Tissue	0.117	lower leg blood vessel	0.719	Other Anatomic Concept
0.117	brachial artery	0.225	Muscle	0.117	abdomen blood vessel	0.649	Skeletal System Part
0.117	celiac artery	0.223	Cardiovascular System Part	0.117	pelvis blood vessel	0.521	Respiratory System

Table 13. Top nine aggregate anchor distance discrepancies $D(a)$ for DSSIM and Taxomap alignments.

As part of a broader infrastructure for the analytical management of ontologies and alignments, further development of these methods is required. Nonetheless, these results suggest that minimizing discrepancy may be related to alignment quality. Thus discrepancy may be an important adjunct to alignment evaluation, playing a role as an automatic pre-filter for hand-built alignments. Moreover, the detailed examination of how particular links and anchors participate with respect to discrepancy within an overall alignment should have high utility for knowledge managers and ontology engineers, revealing details of the nature and structure of the mappings being considered. Perhaps most exciting is the dual problem to that considered here: given an alignment F which is *a priori* believed to be of high quality, how can $D(F)$ be used to aid in the *design* of those ontologies? Some of the results above are very suggestive of these possibilities.

5 Acknowledgements

Thanks to Sinan al-Saffar and a number of reviewers for their assistance in improving a prior version of this paper. Much thanks to Christian Meilicke at Universität Mannheim for extensive consultation about the OAEI-2008 Anatomy track. Thanks also to Martin Ringwald and Terry Hayamizu of the Jackson Laboratory for allowing us to access the full reference alignment for the Anatomy track of OAEI-2009. Joshua Short at PNNL also assisted with a number of things.

References

1. Butanitsky, A and Hirst, G: (2006) "Evaluating WordNet-based Measures of Lexical Semantic Relatedness", *Computational Linguistics*, v. **32**:1, pp. 13-47
2. Caracciolo, Caterina; Stuckenschmidt, Heiner; Svab, Ondrej; *et al.*: (2008) "First Results of the Ontology Alignment Evaluation Initiative 2008", in: *Proc. 3rd Int. Wshop. On Ontology Matching (OM2008)*,
3. Davey, BA and Priestly, HA: (1990) *Introduction to Lattices and Order*, Cambridge UP, Cambridge UK, 2nd Edition
4. Euzenat, J and Shvaiko, P: (2007) *Ontology Matching*, Springer-Verlag, Hiedelberg
5. Falconer, Sean M and Maslov, Dmitri: (2006) "Hierarchical Alignment of Weighted Directed Acyclic Graphs", `arVix:cs.DS/0606124v1`
6. Fellbaum, Christiane, ed.: (1998) *Wordnet: An Electronic Lexical Database*, MIT Press, Cambridge, MA
7. Feng, Y; Goldstone, RL; and Menkov, V: (2004) "ABSURDIST II: A Graph Matching Algorithm and its Application to Conceptual System Translation", in: *Proc. 7th Int. Florida AI Research Society Conference (FLAIRS 04)*, v. **2**, pp. 640-645
8. He, Hu and Xiaoyong, Du: (2007) "Ontology Hierarchies Matching by Lattices Alignment", in: *Proc. Ontology Matching 2007 (OM-2007), ISWC 2007*
9. Joslyn, Cliff: (2009) "Hierarchy Analysis of Knowledge Networks", in: *IJCAI Int. Wshop. on Graph Structures for Knowledge Representation and Reasoning*, in press
10. Joslyn, Cliff; Baddeley, Bob; Blake, Judith; *et al.*: (2009) "Automated Annotation-Based Bio-Ontology Alignment with Structural Validation", in: *Proc. Int. Conf. on Biomedical Ontology (ICBO 09)*
11. Joslyn, Cliff; Donaldson, Alex; and Paulson, Patrick: (2008) "Evaluating the Structural Quality of Semantic Hierarchy Alignments", poster at the *Int. Semantic Web Conf. (ISWC 08)*, `http://dblp.uni-trier.de/db/conf/semweb/iswc2008p.html#JoslynDP08`
12. Y Kalfoglou, M Schorlemmer: (2002) "IF-Map: An Ontology-Mapping Method based on Information-Flow Theory", Proc. 1st Int. Conf. Ontologies, Databases and Application of Semantics (ODBASE'02), Irvine, CA, USA
13. Lord, PW; Stevens, Robert; Brass, A; CA Goble: (2003) "Investigating Semantic Similarity Measures Across the Gene Ontology: the Relationship Between Sequence and Annotation", *Bioinformatics*, v. **10**, pp. 1275-1283
14. Maedche, Alexander and Staab, Steffen: (2002) "Measuring Similarity Between Ontologies", in: *Proc. 13th Int. Conf. Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, LNCS*, v. **2473**, pp. 251-263
15. Meilicke, Christian and Stuckenschmidt, Heiner: (2008) "Incoherence as a Basis for Measuring the Quality of Ontology Mappings", in: *Proc. 3rd Int. Wshop On Ontology Matching (OM2008)*, ed. Pavel Shvaiko et al., Karlsruhe
16. Monjardet, B: (1981) "Metrics on Partially Ordered Sets - A Survey", *Discrete Mathematics*, v. **35**, pp. 173-184
17. Noy, N and Musen, MA: (2003) "The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping", *Int. J. Human-Computer Studies*, v. **59**, pp. 983-1024
18. Orum, C and Joslyn, CA: (2009) "Valuations and Metrics on Partially Ordered Sets", `http://arxiv.org/abs/0903.2679v1`, submitted
19. Sanfilippo, A; Posse, C; Gopalan, B; *et al.*: (2007) "Combining Hierarchical and Associative Gene Ontology Relations With Textual Evidence in Estimating Gene and Gene Product Similarity", *IEEE Trans. on Nanobioscience*, v. **6**:1, pp. 51-59
20. Schröder, Bernd SW: (2003) *Ordered Sets*, Birkhauser, Boston

Results of the Ontology Alignment Evaluation Initiative 2009*

Jérôme Euzenat[1], Alfio Ferrara[7], Laura Hollink[2], Antoine Isaac[2], Cliff Joslyn[10], Véronique Malaisé[2], Christian Meilicke[3], Andriy Nikolov[8], Juan Pane[4], Marta Sabou[8], François Scharffe[1], Pavel Shvaiko[5], Vassilis Spiliopoulos[9], Heiner Stuckenschmidt[3], Ondřej Šváb-Zamazal[6], Vojtěch Svátek[6], Cássia Trojahn[1], George Vouros[9], and Shenghui Wang[2]

[1] INRIA & LIG, Montbonnot, France
{Jerome.Euzenat,Francois.Scharffe,Cassia.Trojahn}@inrialpes.fr
[2] Vrije Universiteit Amsterdam, The Netherlands
{laurah,vmalaise,aisaac,swang}@few.vu.nl
[3] University of Mannheim, Mannheim, Germany
{christian,heiner}@informatik.uni-mannheim.de
[4] University of Trento, Povo, Trento, Italy
pane@dit.unitn.it
[5] TasLab, Informatica Trentina, Trento, Italy
pavel.shvaiko@infotn.it
[6] University of Economics, Prague, Czech Republic
{svabo,svatek}@vse.cz
[7] Universita degli studi di Milano, Italy
ferrara@dico.unimi.it
[8] The Open university, UK
{r.sabou,a.nikolov}@open.ac.uk
[9] University of the Aegean, Greece
{vspiliop,georgev}@aegean.gr
[10] Pacific Northwest National Laboratory, USA
cliff.joslyn@pnl.gov

Abstract. Ontology matching consists of finding correspondences between ontology entities. OAEI campaigns aim at comparing ontology matching systems on precisely defined test cases. Test cases can use ontologies of different nature (from expressive OWL ontologies to simple directories) and use different modalities, e.g., blind evaluation, open evaluation, consensus. OAEI-2009 builds over previous campaigns by having 5 tracks with 11 test cases followed by 16 participants. This paper is an overall presentation of the OAEI 2009 campaign.

1 Introduction

The Ontology Alignment Evaluation Initiative[1] (OAEI) is a coordinated international initiative that organizes the evaluation of the increasing number of ontology matching

* This paper improves on the "Preliminary results" initially published in the on-site proceedings of the ISWC workshop on Ontology Matching (OM-2009). The only official results of the campaign, however, are on the OAEI web site.

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

systems [10]. The main goal of OAEI is to compare systems and algorithms on the same basis and to allow anyone for drawing conclusions about the best matching strategies. Our ambition is that from such evaluations, tool developers can learn and improve their systems. The OAEI campaign provides the evaluation of matching systems on consensus test cases.

Two first events were organized in 2004: (i) the Information Interpretation and Integration Conference (I3CON) held at the NIST Performance Metrics for Intelligent Systems (PerMIS) workshop and (ii) the Ontology Alignment Contest held at the Evaluation of Ontology-based Tools (EON) workshop of the annual International Semantic Web Conference (ISWC) [23]. Then, unique OAEI campaigns occurred in 2005 at the workshop on Integrating Ontologies held in conjunction with the International Conference on Knowledge Capture (K-Cap) [2], in 2006 at the first Ontology Matching workshop collocated with ISWC [9], in 2007 at the second Ontology Matching workshop collocated with ISWC+ASWC [11], and in 2008, OAEI results were presented at the third Ontology Matching workshop collocated with ISWC [4]. Finally, in 2009, OAEI results were presented at the fourth Ontology Matching workshop collocated with ISWC, in Chantilly, Virginia USA[2].

We have continued previous years' trend by having a large variety of test cases that emphasize different aspects of ontology matching. This year we introduced two new tracks that have been identified in the previous years:

oriented alignments in which the reference alignments are not restricted to equivalence but also comprise subsumption relations;

instance matching dedicated to the delivery of alignment between instances as necessary for producing linked data.

This paper serves as an introduction to the evaluation campaign of 2009 and to the results provided in the following papers. The remainder of the paper is organized as follows. In Section 2 we present the overall testing methodology that has been used. Sections 3-10 discuss in turn the settings and the results of each of the test cases. Section 11 evaluates, across all tracks, the participant results with respect to their capacity to preserve the structure of ontologies. Section 12 overviews lessons learned from the campaign. Finally, Section 13 outlines future plans and Section 14 concludes the paper.

2 General methodology

We first present the test cases proposed this year to OAEI participants. Then, we describe the three steps of the OAEI campaign and report on the general execution of the campaign. In particular, we list participants and the tests they considered.

2.1 Tracks and test cases

This year's campaign has consisted of 5 tracks gathering 11 data sets and different evaluation modalities.

[2] http://om2009.ontologymatching.org

The benchmark track (§3): Like in previous campaigns, a systematic benchmark series has been produced. The goal of this benchmark series is to identify the areas in which each matching algorithm is strong and weak. The test is based on one particular ontology dedicated to the very narrow domain of bibliography and a number of alternative ontologies of the same domain for which alignments are provided.

The expressive ontologies track offers ontologies using OWL modeling capabilities:

Anatomy (§4): The anatomy real world case is about matching the Adult Mouse Anatomy (2744 classes) and the NCI Thesaurus (3304 classes) describing the human anatomy.

Conference (§5): Participants are asked to find all correct correspondences (equivalence and/or subsumption) and/or 'interesting correspondences' within a collection of ontologies describing the domain of organizing conferences (the domain being well understandable for every researcher). Results are evaluated a posteriori in part manually and in part by data-mining techniques and logical reasoning techniques. They are also evaluated against reference alignments based on a subset of the whole collection.

The directories and thesauri track proposes web directories, thesauri and generally less expressive resources:

Fishery gears: This test case features four different classification schemes, expressed in OWL, adopted by different fishery information systems in FIM division of FAO. An alignment performed on this 4 schemes should be able to spot out equivalence, or a degree of similarity between the fishing gear types and the groups of gears, so as to enable a future exercise of data aggregation across systems.

Directory (§6): The directory real world case consists of matching web sites directories (like open directory or Yahoo's). It is more than 4 thousand elementary tests.

Library (§7): Three large SKOS subject heading lists for libraries have to be matched using relations from the SKOS vocabulary. Results are evaluated on the basis of (i) a partial reference alignment (ii) using the alignments to reindex books from one vocabulary to the other.

Oriented alignments (benchmark-subs §8) :

This track focuses on the evaluation of alignments that contain other relations than equivalences.

Instance matching (§9): The instance data matching track aims at evaluating tools able to identify similar instances among different datasets. It features Web datasets, as well as a generated benchmark:

Eprints-Rexa-Sweto/DBLP benchmark (ARS) three datasets containing instances from the domain of scientific publications;

TAP-Sweto-Tesped-DBpedia three datasets covering several topics and structured according to different ontologies;

IIMB A benchmark generated using one dataset and modifying it according to various criteria.

Very large crosslingual resources (§10): The purpose of this task (vlcr) is to match the Thesaurus of the Netherlands Institute for Sound and Vision (called GTAA) to two other resources: the English WordNet from Princeton University and DBpedia.

Table 1 summarizes the variation in the results expected from these tests.

For the first time this year we had to cancel two tracks, namely Fishery and TAP-Sweto-Tesped-DBpedia due to the lack of participants. This is a pity for those who have prepared these tracks, and we will investigate what led to this situation in order to improve next year.

test	formalism	relations	confidence	modalities	language
benchmarks	OWL	=	[0 1]	open	EN
anatomy	OWL	=	[0 1]	blind	EN
conference	OWL-DL	=, <=	[0 1]	blind+open	EN
fishery	OWL	=	1	expert	EN+FR+ES
directory	OWL	=	1	blind+open	EN
library	SKOS	exact-,narrow-,	1	blind	EN+DU+FR
	+OWL	broadMatch			
benchmarksubs	OWL	=,<,>	[0 1]	open	EN
ars	RDF	=	[0 1]	open	EN
tap	RDF	=	[0 1]	open	EN
iimb	RDF	=	[0 1]	open	EN
vlcr	SKOS	exact-,	[0 1]	blind	DU+EN
	+OWL	closeMatch		expert	

Table 1. Characteristics of test cases (open evaluation is made with already published reference alignments, blind evaluation is made by organizers from reference alignments unknown to the participants, consensual evaluation is obtained by reaching consensus over the found results).

2.2 Preparatory phase

Ontologies to be matched and (where applicable) reference alignments have been provided in advance during the period between June 1^{st} and June 22^{nd}, 2009. This gave potential participants the occasion to send observations, bug corrections, remarks and other test cases to the organizers. The goal of this preparatory period is to ensure that the delivered tests make sense to the participants. The final test base was released on July 6th. The data sets did not evolve after this period.

2.3 Execution phase

During the execution phase, participants used their systems to automatically match the ontologies from the test cases. Participants have been asked to use one algorithm and the same set of parameters for all tests in all tracks. It is fair to select the set of parameters that provide the best results (for the tests where results are known). Beside parameters, the input of the algorithms must be the two ontologies to be matched and any general purpose resource available to everyone, i.e., no resource especially designed for the test. In particular, participants should not use the data (ontologies and reference alignments) from other test cases to help their algorithms. In most cases, ontologies are described in OWL-DL and serialized in the RDF/XML format. The expected alignments are provided in the Alignment format expressed in RDF/XML [8]. Participants also provided the papers that are published hereafter and a link to their systems and their configuration parameters.

2.4 Evaluation phase

The organizers have evaluated the alignments provided by the participants and returned comparisons on these results.

In order to ensure that it is possible to process automatically the provided results, the participants have been requested to provide (preliminary) results by September 1^{st}. In the case of blind tests only the organizers did the evaluation with regard to the withheld reference alignments.

The standard evaluation measures are precision and recall computed against the reference alignments. For the matter of aggregation of the measures we use weighted harmonic means (weights being the size of the true positives). This clearly helps in the case of empty alignments. Another technique that has been used is the computation of precision/recall graphs so it was advised that participants provide their results with a weight to each correspondence they found. New measures addressing some limitations of precision and recall have also been used for testing purposes as well as measures compensating for the lack of complete reference alignments.

2.5 Comments on the execution

After a decreased number of participants last year, this year the number increased again: 4 participants in 2004, 7 in 2005, 10 in 2006, 17 in 2007, 13 in 2008, and 16 in 2009.

The number of covered runs has slightly increased: 53 in 2009, 50 in 2008, and 48 in 2007. This may be due to the increasing specialization of tests: some systems are specifically designed for instance matching or for anatomy.

We have had not enough time to systematically validate the results which had been provided by the participants, but we run a few systems and we scrutinized some of the results.

The list of participants is summarized in Table 2. Similar to previous years not all participants provided results for all tests. They usually did those which are easier to run, such as benchmark, anatomy, directory, and conference. The variety of tests and the short time given to provide results have certainly prevented participants from considering more tests.

The sets of participants is divided in two main categories: those who participated in the instance matching track and those who participated in ontology matching tracks. Only a few systems (DSSim and RiMOM) participated in both types of tracks.

The summary of the results track by track is provided in the following sections.

3 Benchmark

The goal of the benchmark tests is to provide a stable and detailed picture of each algorithm. For that purpose, the algorithms are run on systematically generated test cases.

System	aflood	AgrMaker	AMExt	AROMA	ASMOV	DSSim	FBEM	GeRoMe	GG2WW	HMatch	kosimap	Lily	MapPSO	RiMOM	SOBOM	TaxoMap	Total=16
Confidence		√	√	√	√	√	√	√	√	√	√	√		√	√	√	
benchmarks	√	√		√	√	√		√			√	√	√	√	√	√	12
anatomy	√	√		√	√	√					√	√		√	√	√	10
conference	√	√	√	√	√	√					√						7
directory	√				√	√					√	√			√	√	7
library																√	1
benchmarksubs					√									√		√	3
ars					√	√	√			√				√			5
iimb	√				√	√	√			√				√			6
vlcr						√			√								2
Total	5	3	1	3	7	7	2	1	1	2	4	3	1	5	3	5	53

Table 2. Participants and the state of their submissions. Confidence stands for the type of result returned by a system: it is ticked when the confidence has been measured as non boolean value.

3.1 Test data

The domain of this first test is Bibliographic references. It is based on a subjective view of what must be a bibliographic ontology. There may be many different classifications of publications, for example, based on area and quality. The one chosen here is common among scholars and is based on publication categories; as many ontologies (tests #301-304), it is reminiscent to BibTeX.

The systematic benchmark test set is built around one reference ontology and many variations of it. The ontologies are described in OWL-DL and serialized in the RDF/XML format. The reference ontology is that of test #101. It contains 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals. Participants have to match this reference ontology with the variations. Variations are focused on the characterization of the behavior of the tools rather than having them compete on real-life problems. They are organized in three groups:

Simple tests (1xx) such as comparing the reference ontology with itself, with another irrelevant ontology (the wine ontology used in the OWL primer) or the same ontology in its restriction to OWL-Lite;

Systematic tests (2xx) obtained by discarding features from some reference ontology. It aims at evaluating how an algorithm behaves when a particular type of information is lacking. The considered features were:

- *Name of entities* that can be replaced by random strings, synonyms, name with different conventions, strings in another language than English;
- *Comments* that can be suppressed or translated in another language;
- *Specialization hierarchy* that can be suppressed, expanded or flattened;
- *Instances* that can be suppressed;
- *Properties* that can be suppressed or having the restrictions on classes discarded;

– *Classes* that can be expanded, i.e., replaced by several classes or flattened.

Four real-life ontologies of bibliographic references (3xx) found on the web and left mostly untouched (there were added xmlns and xml:base attributes).

Since the goal of these tests is to offer some kind of permanent benchmarks to be used by many, the test is an extension of the 2004 EON Ontology Alignment Contest, whose test numbering it (almost) fully preserves.

The tests are roughly the same as last year. We only suppressed some correspondences that rendered the merged ontologies inconsistent (in 301 and 304) since an increasing number of systems were able to test the consistency of the resulting alignments.

The kind of expected alignments is still limited: they only match named classes and properties, they mostly use the "=" relation with confidence of 1. Full description of these tests can be found on the OAEI web site.

3.2 Results

Twelve systems participated in the benchmark track of this year's campaign (see Table 2). Three systems that had participated last year (CIDER, SAMBO, and SPIDER) did not participate this year.

Table 3 shows the results, by groups of tests. The results of last year are also provided. We display the results of participants as well as those given by some simple edit distance algorithm on labels (edna). The computed values are real precision and recall and not an average of precision and recall. The full results are on the OAEI web site.

As shown in Table 3, two systems are ahead: Lily and ASMOV, with aflood and RiMOM as close followers (with GeRoME, AROMA, DSSim, and AgreementMaker – which is referred as AgrMaker in the tables and figures – having intermediary performance). Last year, ASMOV, Lily and RiMOM had the best performance, followed by AROMA, DSSim, and aflood. No system had strictly lower performance than edna.

Looking for each group of tests, in simple tests (1xx) all systems have similar performance, excluding SOBOM and TaxoMap. Each algorithm has its best score with the 1xx test series. For systematic tests (2xx), which allows to distinguish the strengths of algorithms, Lily and ASMOV are again ahead of the other systems. Finally, for real cases (3xx), AgreementMaker and aflood provide the best results, with Lily, RiMOM, ASMOV, AROMA, and DSSim as followers. There is no a unique best system for all group cases.

Looking for improvements in the systems participating both this year and in the last campaign, GeRoMe and MapPSO have significantly improved their results both in terms of precision and recall, while aflood provides better recall and AROMA improves its results in real cases.

The results have also been compared with the symmetric measure proposed in [7]. It is a generalisation of precision and recall in order to better discriminate systems that slightly miss the target from those which are grossly wrong. This measure slightly improves traditional precision and recall, which are displayed in Table 3 ("Symmetric relaxed measures"). This year, MapPSO has significantly better symmetric precision and recall than classical precision and recall, to the point that it is at the level of the best

system	refalign		edna		aflood		AgrMaker		AROMA		ASMOV		DSSim		GeRoMe		kosimap		Lily		MapPSO		RiMOM		SOBOM		TaxoMap	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
2009																												
1xx	1.00	1.00	0.96	1.00	1.00	1.00	0.98	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.97	1.00	0.34
2xx	1.00	1.00	0.41	0.56	0.98	0.74	0.98	0.60	0.98	0.69	0.96	0.85	0.97	0.62	0.92	0.71	0.94	0.57	0.97	0.86	0.73	0.73	0.93	0.81	0.97	0.46	0.90	0.23
3xx	1.00	1.00	0.47	0.82	0.90	0.81	0.92	0.79	0.85	0.78	0.81	0.82	0.94	0.67	0.68	0.60	0.72	0.50	0.84	0.81	0.54	0.29	0.81	0.82	0.92	0.55	0.77	0.31
H-mean	1.00	1.00	0.43	0.59	0.98	0.80	0.99	0.62	0.94	0.69	0.95	0.87	0.97	0.66	0.91	0.73	0.91	0.59	0.97	0.88	0.63	0.61	0.93	0.82	0.98	0.44	0.86	0.26
Symmetric relaxed measures																												
H-mean	1.00	1.00	0.73	1.00	0.99	0.81	0.99	0.62	0.98	0.72	0.99	0.90	1.00	0.67	0.92	0.74	0.99	0.64	0.99	0.89	0.99	0.96	0.99	0.88	1.00	0.44	0.99	0.30
2008																												
1xx	1.00	1.00	0.96	1.00	1.00	1.00			1.00	1.00	1.00	1.00	1.00	1.00	0.96	0.79			1.00	1.00	0.92	1.00	1.00	1.00			1.00	0.34
2xx	1.00	1.00	0.41	0.56	0.96	0.69			0.96	0.70	0.95	0.85	0.97	0.64	0.56	0.52			0.97	0.86	0.48	0.53	0.96	0.82			0.95	0.21
3xx	1.00	1.00	0.47	0.82	0.95	0.66			0.82	0.71	0.81	0.77	0.90	0.71	0.61	0.40			0.87	0.81	0.49	0.25	0.80	0.81			0.92	0.21
H-mean	1.00	1.00	0.43	0.59	0.97	0.71			0.95	0.70	0.95	0.86	0.97	0.67	0.60	0.58			0.97	0.88	0.51	0.54	0.96	0.84			0.91	0.22

Table 3. Means of results obtained by participants on the benchmark test case (corresponding to harmonic means). The symmetric relaxed measure corresponds to the relaxed precision and recall measures of [7].

systems. This may be due the kind of algorithm which is used, that misses the target, but not by far.

Figure 2 shows the precision and recall graphs of this year. These results are only relevant for the results of participants who provide confidence measures different from 1 or 0 (see Table 2). This graph has been drawn with only technical adaptation of the technique used in TREC. Moreover, due to lack of time, these graphs have been computed by averaging the graphs of each of the tests (instead to pure precision and recall).

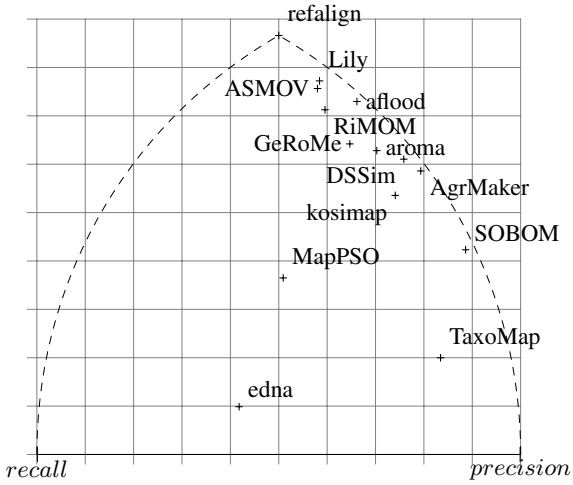


Fig. 1. Each point expresses the position of a system with regard to precision and recall. This shows that most of the systems favor precision over recall.

These results and those displayed in Figure 1 single out the same group of systems, Lily, ASMOV, aflood, and RiMOM which seem to perform these tests at the highest level of quality. Of these, Lily and ASMOV have slightly better results than the two others. So, this confirms the leadership that we observed on raw results.

Like in the three previous campaigns, there is a gap between these systems and their followers (GeRoME, AROMA, DSSim, and AgreementMaker).

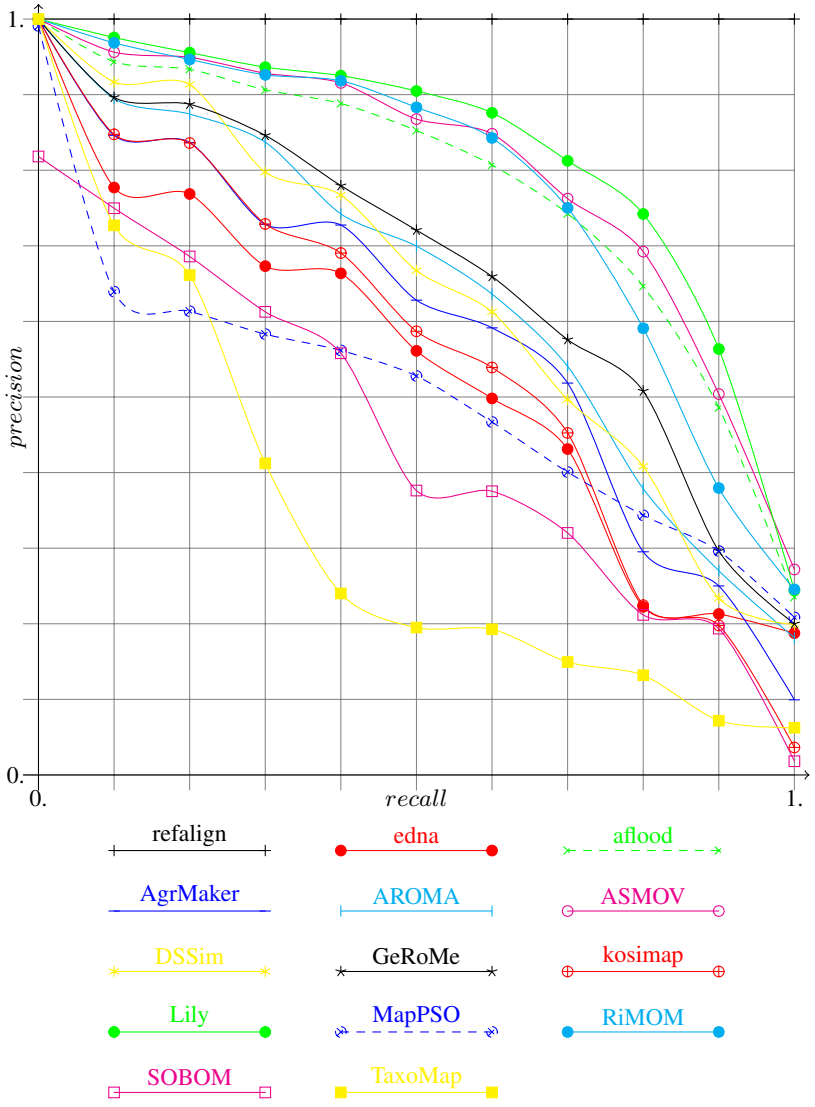


Fig. 2. Precision/recall graphs for benchmarks. The results given by the participants are cut under a threshold necessary for achieving $n\%$ recall and the corresponding precision is computed. Systems for which these graphs are not meaningful (because they did not provide graded confidence values) are drawn in dashed lines.

4 Anatomy

Within the anatomy track we confront existing matching technology with real world ontologies. Currently, we find such real world cases primarily in the biomedical domain, where a significant number of ontologies have been built covering different aspects of medical research. Due to the complexity and the specialized vocabulary of the domain, matching biomedical ontologies is one of the hardest alignment problems.

4.1 Test data and experimental setting

The ontologies of the anatomy track are the NCI Thesaurus describing the human anatomy, published by the National Cancer Institute (NCI)[3], and the Adult Mouse Anatomical Dictionary[4], which has been developed as part of the Mouse Gene Expression Database project. Both resources are part of the Open Biomedical Ontologies (OBO). A detailed description of the data set has been given in the context of OAEI 2007 [11] and 2008 [4].

As proposed in 2008 the task of automatically generating an alignment has been divided into four subtasks. Task #1 is obligatory for participants of the anatomy track, while task #2, #3 and #4 are optional tasks.

- For task #1 the matcher has to be applied with standard settings to obtain a result that is as good as possible with respect to the expected F-measure.
- In task #2 / #3 an alignment has to be generated that favors precision over recall and vice versa. Systems configurable with respect to these requirements will be more useful in particular application scenarios.
- In task #4 we simulate that a group of domain experts created an incomplete reference alignment R_p. Given both ontologies as well as R_p, a matching system should be able to exploit the additional information encoded in R_p.

Due to the harmonization of the ontologies applied in the process of generating a reference alignment (see [3] and [11]), a high number of rather trivial correspondences (61%) can be found by simple string comparison techniques. At the same time, we have a good share of non-trivial correspondences (39%). The partial reference alignment used in subtrack #4 is the union of all trivial correspondences and 54 non-trivial correspondences.

4.2 Results

In total, ten systems participated in the anatomy track (in 2007 there were eleven participants, in 2008 nine systems participated). An overview is given in Table 4. While the number of participants is stable, we find systems participating for the first time (SOBOM, kosimap), systems re-entering the competition after a year of absence (AgreementMaker, which is referred to as AgrMaker in the tables) and systems continuously participating (ASMOV, DSSim, Lily, RiMOM, TaxoMap).

[3] http://www.cancer.gov/cancerinfo/terminologyresources/

[4] http://www.informatics.jax.org/searches/AMA_form.shtml

System	2007	2008	2009
aflood	-	√	√
AgrMaker	√	-	√+
AROMA	-	√	√
AOAS	√+	-	-
ASMOV	√	√	√
DSSim	√	√	√
Falcon-AO	√	-	-
kosimap	-	-	√
Lily	√	√	√
Prior+	√	-	-
RiMOM	√	√+	√
SAMBO	√+	√+	-
SOBOM	-	-	√+
TaxoMap	√	√	√
X-SOM	√	-	-
avg. F-measure	0.598	0.718	0.764

Table 4. Overview on anatomy participants from 2007 to 2009, a √-symbol indicates that the system participated, + indicates that the system achieved an F-measure ≥ 0.8 in subtrack #1.

In Table 4 we have marked the participants with an F-measure ≥ 0.8 with a + symbol. Unfortunately, the top performers of the last two years do not participate this year (AOAS in 2007, SAMBO in 2008). In the last row of the table the average of the obtained F-measures is shown. We observe significant improvements over time. However, in each of the three years the top systems generated alignments with F-measure of ≈ 0.85. It seems that there is an upper bound which is hard to exceed.

Runtime Due to the evaluation process of the OAEI, the submitted alignments have been generated by the participants, who run the respective systems on their own machines. Nevertheless, the resulting runtime measurements provide an approximate basis for a useful comparison. In 2007, we observed significant differences with respect to the stated runtimes. Lily required several days for completing the matching task and more than half of the systems could not match the ontologies in less than one hour. In 2008 we already observed increased runtimes. This year's evaluation revealed that only one system still requires more than one hour. The fastest system is aflood (15 sec) followed by AROMA, which requires approximately 1 minute. Notice that aflood is run with a configuration optimized for runtime efficiency in task #1, it requires 4 minutes with a configuration which aims at generating an optimal alignment used for #2, #3, and #4. Detailed information about runtimes can be found in the second column of Table 5.

Results for subtracks #1, #2 and #3 Table 5 lists the results of the participants in descending order with respect to the F-measure achieved for subtrack #1. In the first two rows we find SOBOM and AgreementMaker. Both systems have very good results and distance themselves from the remaining systems. SOBOM, although participating for the first time, submitted the best result in 2009. The system seems to be optimized

System	Task #1				Task #2			Task #3			Recall+	
	Runtime	Prec.	Rec.	F	Prec.	Rec.	F	Prec.	Rec.	F	#1	#3
SOBOM	≈ 19 min	0.952	0.777	0.855	-	-	-	-	-	-	0.431	-
AgrMaker	≈ 23 min	0.865	0.798	0.831	0.967	0.682	0.800	0.511	0.815	0.628	0.489	0.553
RiMOM	≈ 10 min	0.940	0.684	0.792	-	-	-	-	-	-	0.183	-
TaxoMap	≈ 12 min	0.870	0.678	0.762	0.953	0.609	0.743	0.458	0.716	0.559	0.222	0.319
DSSim	≈ 12 min	0.853	0.676	0.754	0.973	0.620	0.757	0.041	0.135	0.063	0.185	0.061
ASMOV	≈ 5 min	0.746	0.755	0.751	0.821	0.736	0.776	0.725	0.767	0.745	0.419	0.474
aflood	≈ 15 sec / 4 min	0.873	0.653	0.747	0.892	0.712	0.792	0.827	0.763	0.794	0.197	0.484
Lily	≈ 99 min	0.738	0.739	0.739	0.869	0.559	0.681	0.534	0.774	0.632	0.477	0.548
AROMA	≈ 1 min	0.775	0.678	0.723	-	-	-	-	-	-	0.368	-
kosimap	≈ 5 min	0.866	0.619	0.722	0.907	0.446	0.598	0.866	0.619	0.722	0.154	0.154

Table 5. Participants and results with respect to runtime, precision, recall, recall+ and F-measure.

for generating a precise alignment, however, the submitted alignment contains also a number of non trivial correspondences (see the column Recall+ for subtrack #1).[5]

AgreementMaker generates a less precise alignment, but manages to output a higher number of correct correspondences. None of the other systems detected a higher number of non-trivial correspondences for both subtrack #1 and #3 in 2009. However, it cannot top the SAMBO submission of 2008, which is known for its extensive use of biomedical background knowledge.

The RiMOM system is slightly worse with respect to the achieved F-measure compared to its 2008 submission. The precision has been improved, however, this caused a loss of recall and in particular a significant loss of recall+. Unfortunately, RiMOM did not participate in subtask #3, so we cannot make statements about its strength in detecting non-trivial correspondences based on a different configuration.

The systems listed in the following columns achieve similar results with respect to the overall quality of the generated alignments (F-measures between 0.72 and 0.76). However, significant differences can be found in terms of the trade-off between precision and recall. All systems except ASMOV and Lily favor precision over recall. Notice that a F-measure of 0.755 can easily be achieved by constructing a highly precise alignment without detecting any non-trivial correspondences. At the same time it is relatively hard to generate an alignment with a F-measure of 0.755 that favors recall over precision. Thus, the results of ASMOV and Lily have to be interpreted more positively than indicated by the F-measure.

The observation that it is not hard to construct a highly precise alignment with acceptable recall is supported by the results of subtask #2, where we find relatively similar results for all participants. In particular, it turned out that some systems (ASMOV, DSSim) have their best F-measure in track #2. The evaluation results for aflood require some additional explanations. aflood is run for track #1 with a configuration which results in a significant reduction of the runtime (15 sec), while for track #2 and #3 the

[5] Recall+ is defined as recall restricted to the subset of non trivial correspondences in the reference alignment. A detailed definition can be found in the results paper of 2007 [11].

system required approximately 4 minutes due to different settings. Therefore, aflood creates better alignments as solutions to subtask #2 and #3.

In 2007 we were surprised by the good performance of the naive label comparison approach. Again, we have to emphasize that this is to a large degree based on the harmonization of the ontologies that has been applied in the context of generating the reference alignment. Nevertheless, the majority of participants was able to top the results of the trivial string matching approach this year.

Results for subtrack #4 In the following we refer to an alignment generated for task #1 resp. #4 as A_1 resp. A_4. This year we have chosen an evaluation strategy that differs from the approach of the last year. We compare $A_1 \cup R_p$ resp. $A_4 \cup R_p$ with the reference alignment R. Thus, we compare the situation where the partial reference alignment is added after the matching process has been conducted against the situation where the partial reference alignment is available as additional resource used within the matching process. The results are presented in Table 6.

System	Δ-Precision	Δ-Recall	Δ-F-Measure
SAMBOdtf$_{2008}$	$+0.020_{\ 0.837 \to 0.856}$	$+0.003_{\ 0.867 \to 0.870}$	$+0.011_{\ 0.852 \to 0.863}$
ASMOV	$+0.034_{\ 0.759 \to 0.792}$	$-0.018_{\ 0.808 \to 0.790}$	$+0.009_{\ 0.782 \to 0.791}$
aflood$_{\#3}$	$+0.005_{\ 0.838 \to 0.843}$	$+0.003_{\ 0.825 \to 0.827}$	$+0.004_{\ 0.831 \to 0.835}$
TaxoMap	$+0.019_{\ 0.878 \to 0.897}$	$-0.026_{\ 0.732 \to 0.706}$	$-0.008_{\ 0.798 \to 0.790}$
AgrMaker	$+0.128_{\ 0.870 \to 0.998}$	$-0.181_{\ 0.831 \to 0.650}$	$-0.063_{\ 0.850 \to 0.787}$

Table 6. Changes in precision, recall and F-measure based on comparing $A_1 \cup R_p$, resp. $A_4 \cup R_p$, against reference alignment R.

Four systems participated in task #4. These systems were aflood, AgreementMaker, ASMOV and TaxoMap. In Table 6 we additionally added a row that displays the 2008 submission of SAMBOdtf, which had the best results for subtrack #4 in 2008. For aflood we used A_3 instead of A_1 to allow a fair comparison, due to the fact that A_1 was generated with runtime optimization configuration.

A first look at the results shows that all systems use the partial reference alignment to increase the precision of their systems. Most of them them have slightly better values for precision (between 0.5% and 3.4%), only AgreementMaker uses the additional information in a way which has a stronger impact in terms of a significantly increased precision. However, only three correspondences have been found that have not been in the partial reference alignment previously[6]. Only SAMBOdtf and aflood profit from the partial reference alignment by a slightly increased recall, while the other systems wrongly filter out some correct correspondences. This might be based on two specifics of the dataset. On the one hand the major part of the reference alignment consists of trivial correspondences easily detectable by string matching algorithms, while the unknown parts share a different characteristic. Any approach which applies machine learning techniques to learn from the partial reference alignment is thus bound to fail. On the other hand parts of the matched ontologies are incomplete with respect to

[6] Notice that we only take correspondences between anatomical concepts into account.

subsumption axioms. As pointed out in [16], the completeness of the structure and the correct use of the structural relations within the ontologies has an important influence on the quality of the results. For these reasons it is extremely hard to use the partial reference alignment in an appropriate way in subtask #4.

4.3 Conclusions

Although it is argued that domain related background knowledge is a crucial point in matching biomedical ontologies (see for example [1; 20]), the results of 2009 raise some doubts about this issue. While in 2007 and 2008 the competition was clearly dominated by matching systems heavily exploiting background knowledge (UMLS), this years top performer SOBOM uses none of these techniques. However, the strong F-measure of SOBOM is mainly based on high precision. Comparing the alignments generated by SAMBO in 2008 and SOBOM in 2009 it turns out that SAMBO detected 136 correct correspondences not found by SOBOM, while SOBOM finds 36 correct correspondences not detected by SAMBO. Unfortunately, SOBOM did not participate in subtrack #3. Thus, it is hard to assess its capability for detecting non-trivial correspondences. The results of subtask #4 are disappointing at first sight. Since this kind of task has been introduced in 2008, we expected better results in 2009. However, it turned out again that only minor positive effects can be achieved. But, as already argued, the task of subtrack #4 is hard and systems with acceptable results in subtrack #4 might obtain good results under better conditions.

5 Conference

The conference test set introduces matching several more-or-less expressive ontologies. Within this track the results of participants are evaluated using diverse evaluation methods. First, classical evaluation wrt. the *reference alignment* was made, for the ontology pairs where this alignment is available. Second, posterior manual evaluation was made for all ontology pairs using even sampling across all matchers. Third, the complete results were submitted to a data mining tool for discovery of association hypotheses, taking into account specific mapping patterns. Fourth, alignment incoherence was analysed with the help of a logical reasoner.

5.1 Test data

The collection consists of fifteen ontologies in the domain of organizing conferences. Ontologies have been developed within the OntoFarm project[7]. In contrast to last year's conference track, we alsoconsidered subsumption results in evaluation.

The main features of this test set are:

– *Generally understandable domain.* Most ontology engineers are familiar with organizing conferences. Therefore, they can create their own ontologies as well as evaluate the alignment among their concepts with enough erudition.

[7] `http://nb.vse.cz/~svatek/ontofarm.html`

– *Independence of ontologies.* Ontologies were developed independently and based on different resources, they thus capture the issues in organizing conferences from different points of view and with different terminologies.
– *Relative richness in axioms.* Most ontologies were equipped with DL axioms of various kinds, which opens a way to use semantic matchers.

Ontologies differ in numbers of classes, of properties, in their DL expressivity, but also in underlying resources. Ten ontologies are based *on tools* supporting the task of organizing conferences, two are based on experience of people with *personal participation* in conference organization, and three are based on *web pages* of concrete conferences.

Participants were to provide all correct correspondences (equivalence and/or subsumption) and/or "interesting correspondences" within a collection of ontologies describing the domain of organizing conferences.

This year, results of participants are evaluated by four different methods of evaluation: evaluation based on reference alignment, manual labeling, data mining method, and logical reasoning. In addition, we extended the reference alignment from the previous year. Now we have 21 alignments, which correspond to the complete alignment space between 7 ontologies from the data set. Manual evaluation produced statistics such as precision and will also serve as input into evaluation based on data mining and will help in the process of improving and building a reference alignment. Results of participants are checked with regard to their incoherency. These evaluation methods are concisely described at the track result page.

5.2 Results

We had seven participants: aflood, AgreementMaker (AgrMaker), AMExt (an extended version of AgreementMaker), AROMA, ASMOV, DSSim, and kosimap. Here are some basic data, besides evaluations:

– All participants delivered all 105 alignments, except for aflood, which delivered 103 alignments.
– Two participants (ASMOV and DSSim) delivered not only equivalence correspondences but also subsumptions.
– aflood and DSSim matchers delivered "certain" correspondences; other matchers delivered correspondences with confidence values between 0 and 1.

Evaluation based on reference alignment We evaluated the results of participants against a reference alignment. In the case of ASMOV and DSSim we filtered out subsumptions. It includes all pairwise combinations of different 7 ontologies (21 alignments).

In Table 7, there are traditional precision, recall, and F-measure computed for three different thresholds of certainty factor (0.2, 0.5, and 0.7).

For better comparison we established the confidence threshold which provides the highest average F-measure (Table 8). Precision, Recall, and F-measure are given for this optimal confidence threshold. The dependency of F-measure on confidence threshold

	t=0.2			t=0.5			t=0.7		
	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.
aflood	48%	61%	52%	48%	61%	52%	48%	61%	52%
AgrMaker	45%	61%	50%	45%	61%	50%	6%	55%	56%
AMExt	30%	60%	39%	30%	60%	39%	41%	53%	46%
AROMA	37%	49%	41%	38%	49%	42%	40%	19%	25%
ASMOV	58%	40%	47%	22%	3%	4%	5%	1%	1%
DSSim	15%	51%	22%	15%	51%	22%	15%	51%	22%
kosimap	18%	56%	27%	41%	43%	41%	70%	23%	33%

Table 7. Recall, precision and F-measure for three different confidence thresholds.

matcher	confidence threshold	Prec.	Rec.	FMeas.
aflood	*	48%	61%	52%
AgrMaker	0.75	69%	51%	57%
AMExt	0.75	54%	50%	51%
AROMA	0.53	39%	48%	42%
ASMOV	0.23	68%	38%	47%
DSSim	*	15%	51%	22%
kosimap	0.51	52%	42%	45%

Table 8. Confidence threshold, precision and recall for optimal F-measure for each matcher.

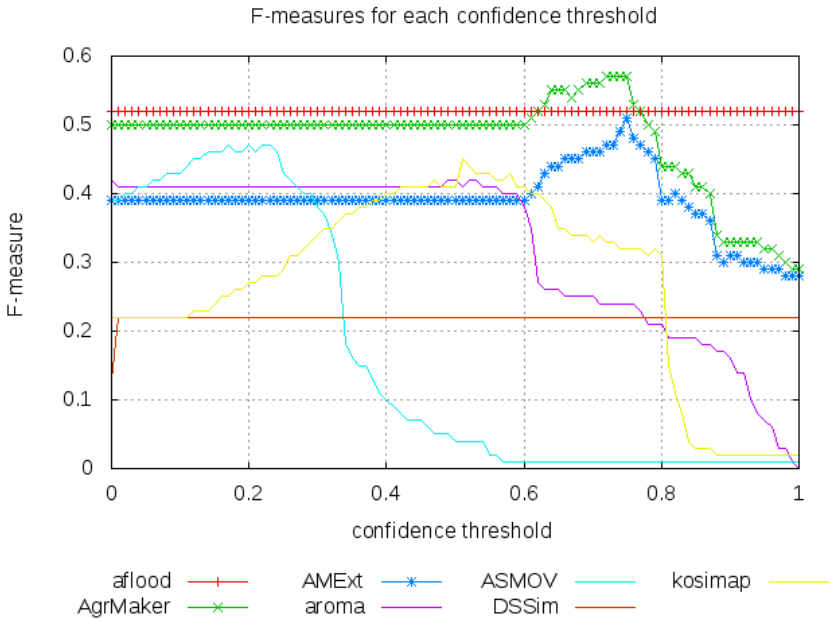


Fig. 3. F-measures depending of confidence.

can be seen from Figure 3. There are two asterisks in the column of confidence threshold for matchers which did not provide graded confidence.

In conclusion, the matcher with the highest average F-measure (.57) is that of AgreementMaker at .75. However we should take into account that this evaluation has been made over small part of all alignments (one fifth).

Comparison with previous year We evaluated the results of participants of OAEI 2008 (ASMOV, DSSim and Lily) against the new reference alignments. For these three matchers from OAEI 2008, we found an optimal confidence threshold in terms of highest average F-measure, see Table 9. In the case of DSSim there is an asterisk because this matcher did not provide graded confidence.

In conclusion, the matcher with the highest average F-measure (0.49) was the DSSim. However we should take into account that this evaluation has been made over small part of all alignments (one fifth). We can also compare performance of participants of both years ASMOV and DSSim. While in terms of highest average F-measure ASMOV improved from 43% to 47%, DSSim declined from 49% to 22%. We can also see that ASMOV matcher from OAEI 2009 delivered more correspondences with lower confidence than in OAEI 2008.

matcher	confidence threshold	Prec.	Rec.	FMeas.
ASMOV	0.22	48%	39%	43%
DSSim	*	48%	56%	49%
Lily	0.25	43%	52%	45%

Table 9. Confidence threshold, precision and recall for optimal F-measure for each matcher.

Restricted semantic precision and recall Furthermore, we computed *restricted semantic precision and recall* using a tool from University of Mannheim [12]. We took into account matchers which delivered correspondences with subsumption relations, i.e., ASMOV and DSSim. In Table 10 there are two different semantics variants (natural and pragmatic) of restricted semantic precision and recall computed for confidence threshold 0.23[8].

	natural		pragmatic	
matcher	Prec.	Rec.	Prec.	Rec.
ASMOV	83%	65%	86%	68%
DSSim	1.7%	94%	2%	95%

Table 10. Restricted semantic precision and recall for a confidence threshold of 0.23.

In conclusion, from Table 10 we can see that considering correspondences with subsumption relations ASMOV has better performance in both precision and recall, whereas DSSim has much better recall at expense of lower precision.

[8] This an optimal confidence threshold in terms of highest F-measure for ASMOV. DSSim does not have graded confidence.

Evaluation based on posterior manual labeling This year we take the most secure, i.e., with highest confidence, correct correspondences as a population for each matcher. It means we evaluate 150 correspondences per matcher randomly chosen from all correspondences of all 105 alignments with confidence 1.0 (sampling). Because AROMA, ASMOV and kosimap do not have enough correspondences with 1.0 confidence we take 150 correspondences with highest confidence. In the case of AROMA it was not possible to distinguish between all 153 correspondences so we sampled over its population.

In table 11 you can see approximated precisions for each matcher over its population of best correspondences. N is a population of all the best correspondences for one matcher. n is a number of randomly chosen correspondences so as to have 150 best correspondences for each matcher. TP is a number of correct correspondences from the sample, and P* is an approximation of precision for the correspondences in each population; additionally there is a margin of error computed as: $\frac{\sqrt{(N/n)-1}}{\sqrt{N}}$ based on [24].

matcher	aflood	AgrMaker	AMExt	AROMA	ASMOV	DSSim	kosimap
N	1779	326	360	153	150	5699	150
n	150	150	150	150	150	150	150
TP	74	120	103	83	127	9	144
P*	49%	80%	69%	55%	85%	6%	96%
	±7.8%	±6%	±6.2%	±1.1%		±8.1%	

Table 11. Approximated precision for 150 best correspondences for each matcher.

From table 11 we can conclude that kosimap has the best precision (.96) over its 150 more confident correspondences.

Evaluation based on data mining supported with mapping patterns (based on [19]). As opposed to ontology design patterns[9], which usually concern one ontology, mapping patterns deal with (at least) two ontologies. Mapping patterns reflect the internal structure of ontologies as well as correspondences across the ontologies.

We recognise nine mapping patterns:

- MP1 ("Parent-child triangle"): it consists of an equivalence correspondence between classes A and B and an equivalence correspondence between A and a child of B, where A and B are from different ontologies.
- MP2 ("Mapping along taxonomy"): it consists of simultaneous equivalence correspondences between parents and between children.
- MP3 ("Sibling-sibling triangle"): it consists of simultaneous correspondences between class A and two sibling classes C and D where A is from one ontology and C and D are from another ontology.
- MP4: it is inspired by the 'class-by-attribute' correspondence pattern, where the class in one ontology is restricted to only those instances having a particular value for a a given attribute/relation.

[9] See `http://ontologydesignpatterns.org`.

– MP5: it is inspired by the "composite" correspondence pattern. It consists of a class-to-class equivalence correspondence and a property-to-property equivalence correspondence, where classes from the first correspondence are in the domain or in the range of properties from the second correspondence.
– MP6: it is inspired by the "attribute to relation" correspondence pattern where a datatype and an object property are aligned as an equivalence correspondence.
– MP7: it is the variant of the MP5 "composite pattern". It consists of an equivalence correspondence between two classes and an equivalence correspondence between two properties, where one class from the first correspondence is in the domain and the other class from that correspondence is in the range of equivalent properties, except the case where domain and range is the same class.
– MP8: it consists of an equivalence correspondence between A and B and an equivalence correspondence between a child of A and a parent of B where A and B are from different ontologies. It is sometimes referred to as criss-cross pattern.
– MP9: it is the variant of MP3, where the two sibling classes C and D are disjoint.

MP4, MP5, and MP6 are inspired by correspondence patterns from [21]. In principle, it is not possible to tell which mapping pattern is desirable or not desirable. This must be decided on the basis of an application context or possible alternatives. However, we could roughly say that while MP2 and MP5 seems to be desirable, MP7, MP8, and MP9 indicate incorrect correspondences related to inconsistency.

In Table 12 there are numbers of occurrences of mapping patterns in results of participants of OAEI 2009. We already see that some patterns are more typical for some systems than for other. Proper quantification of this relationship as well as its combination with other characteristics of correspondences is however the task for a mining tool.

System	MP1	MP2	MP3	MP4	MP5	MP6	MP7	MP8	MP9
aflood	0	168	0	272	158	108	6	4	0
AgrMaker	0	127	0	272	81	209	22	2	0
amext	0	128	0	346	112	419	25	4	0
AROMA	238	206	6	442	35	61	13	12	0
asmov	0	350	0	393	0	0	0	0	0
dssim	479	74	964	962	47	410	24	47	295
kosimap	38	233	159	815	392	62	10	4	22

Table 12. Occurrences of mapping patterns in OAEI 2009 results.

For the *data-mining analysis* we employed the *4ft-Miner* procedure of the *LISp-Miner* data mining system[10] for mining of *association rules*. We found several interesting *association hypotheses*: $t1$ to $t6$ are related to confidence or underlying resources of ontologies (see Table 13) and $m1$ to $m10$ are related to mapping patterns (see Table 14). In total there were 21117 correspondences in the data matrix. We can interpret some of these hypotheses as follows:

[10] `http://lispminer.vse.cz/`

	Antecedent				Succedent	Values	
	System	Confidence	Resource1	Resource2	Result	Supp	AvgDff
t1	AgrMaker	> 0.9	*	*	+	0.01	2.876
t2	ASMOV	< 0.3	*	*	+	0.01	2.546
t3	kosimap	$< 0.3; 0.6)$	*	*	+	0.01	2.497
t4	DSSim	*	i	w	-	0.01	2.287
t5	kosimap	$< 0.3; 0.6)$	*	t	+	0.01	2.267
t6	kosimap	*	*	i	-	0.02	1.215

Table 13. Hypotheses for tasks 1 and 2.

	Antecedent		Succedent	Values-
	System	ResultMP	Supp	AvgDff
m1	ASMOV	MP2	0.02	3.418
m2	AROMA	MP1	0.01	2.434
m3	DSSim	MP3	0.05	2.164
m4	AMExt	MP6	0.02	1.481
m5	ASMOV	MP4	0.02	0.874
m6	kosimap	MP5	0.02	0.874
m7	DSSim	MP9	0.01	2.448
m8	DSSim	MP8	0.002	1.386
m9	AgrMaker	MP7	0.001	1.266
m10	AMExt	MP7	0.001	0.879

Table 14. Association Hypotheses related to Mapping Patterns.

– Hypothesis t1: Correspondences that are produced by system AgreementMaker and have high confidence values (higher than 0.9) are by 287%, i.e. almost four times, more often correct than correspondences produced by all systems with all confidence values (on average).
– Hypothesis t4: Correspondences that are produced by system DSSim where ontology 1 is based on expert knowledge and ontology 2 is based on web are by 228%, i.e., more than three times, more often incorrect than correspondences produced by all systems for all types of ontologies (on average).
– Hypothesis m1: Correspondences that are produced by matcher ASMOV are by 341%, i.e., more than four times, more often part of MP2 than correspondences produced by all systems (on average).
– Hypothesis m4: Correspondences that are produced by matcher AMExt are by 148%, i.e., more than twice, more often part of MP6 than correspondences produced by all systems (on average).
– Hypothesis m7: Correspondences that are produced by matcher DSSim are by 244%, i.e., more than three times, more often part of MP9 than correspondences produced by all systems (on average).
– Hypothesis m9: Correspondences that are produced by matcher AgreementMaker are by 126%, i.e., more twice, more often part of MP7 than correspondences produced by all systems (on average).

In conclusion, regarding the first three hypotheses we could say that AgreementMaker is more sure about correspondences with high values than other matchers, ASMOV is suprisingly more correct about correspondences with low confidence values than other matchers and kosimap is more correct for correspondences with medium confidence values. According to next three hypotheses we could say that kosimap works better with ontologies based on tool than web. Further DSSim has problems with aligning "expert' ontologies" and "web" ontologies.

Regarding the three first mapping patterns, ASMOV found MP2, AROMA MP1, and DSSim MP3. Furthermore, AMExt found MP6 as simple correspondence, which is discutable. Maybe it could be better to find instead of datatype property to object property "property-chain" which would allow mapping between datatype property to datatype property via object property as an intermediate mapping element. ASMOV found some correspondences where one class is restricted over certain property's value (MP4) and kosimap found composite pattern (MP5). Finally, some occurrences of the last three mapping patterns were found over the results of DSSim, AgreementMaker, and AMExt. However these related hypotheses had low support except for DSSim and MP9. Anyway we can say that these matchers could be improved if they check the consistency of their results.

Evaluation based on alignment coherence In 2008 we evaluated for the first time the coherence of the submitted alignments. Again, we picked up the same evaluation approach using the maximum cardinality measure m^t_{card} proposed in [17]. The m^t_{card} measure compares the number of correspondences that have to be removed to arrive at a coherent subset against the number of all correspondences in the alignment. The resulting number can be considered as the degree of alignment incoherence. A number

of 0% means, for example, that the alignment is coherent. In particular, we use the pragmatic alignment semantic as defined in [18] to interpret the correspondences of an alignment.

In our experiments we focused on equivalence correspondences and removed subsumption correspondences from the submitted alignments prior to our evaluation. We applied our evaluation approach to the subset of those matching tasks where a reference alignment is available. We used the Pellet reasoner to perform our experiments and excluded the Iasted ontology, which caused reasoning problems in combination with some of the other ontologies.

Results are presented in Table 15. For all systems we used the alignments after applying the optimal confidence threshold (see subscript), and the systems marked with * are those systems that did not deliver a graded confidence. Comparing the corresponding results, the ASMOV system clearly distances itself from the remaining participants. All of the generated alignments were coherent and thus we measured 0% degree of incoherence. However, the thresholded ASMOV alignments contain only few correspondences compared to the alignments of the other systems, which makes it more probable to construct coherent alignments. Thus, we also included the untresholded ASMOV alignments (no subscript) in our analysis: We measured a degree of incoherence of 1.8%, a value that is still significantly lower compared to the other systems. These results also concide with the results presented in Table 14 related to the occurrence of the MP7 to MP9 mapping patterns.

While the verification component built into ASMOV detects most incoherences, none of the other systems uses similar strategies. We have to conclude that logical aspects play only a subordinate role within the approaches implemented in the other matching systems. Additionally, we analyzed what happens when the verification component of ASMOV is turned off.[11] The results are presented in the ASMOV^x row. Notice that the measured values are now similar to the coherence characteristics of the other systems.

In conclusion, these observations also offer an explanation for the significant difference between DSSim and ASMOV with respect to restricted semantic precision and recall (see again Table 10). Computing restricted semantic precision and recall of an alignment A requires to compute the closure of A with respect to derivable subsumption correspondences. Suppose now that A is incoherent and a large fraction of concepts $C_1, ..., C_n$ in O_1 and $D_1, ..., D_m$ in O_2 becomes unsatisfiable. It follows that A entails each correspondence of the type ... $\sqsupseteq C_i$ with $i = 1 \ldots n$, respectively $D_j \sqsubseteq$ ... with $j = 1 \ldots m$. A highly incoherent alignment will thus entail a huge amount of incorrect correspondences. This is the explanation for DSSim's low precision of approximately 2%. These considerations also indicate that the degree of incoherence might have a strong effect on any application that requires to exploit an alignment in a reasoning context.

[11] We would like to thank Yves R. Jean-Mary for providing us with the corresponding set of alignments.

System	Correspondences	Incoherent Alignments	m^t_{card}-mean
ASMOV$_{.23}$	140	0	0.0%
ASMOV	233	3	1.8%
kosimap$_{.51}$	189	6	10.6%
ASMOV^x	316	13	14.7%
AgrMaker$_{.75}$	173	12	15.0%
aflood*	288	15	19.8%
AROMA$_{.53}$	264	13	20.1%
AMExt$_{.75}$	236	13	20.3%
DSSim*	789	15	> 42.2%

Table 15. Number of evaluated correspondences, number of coherent alignments (15 alignments have been analyzed), mean of the maximum cardinality measure. Subscripts refer to the application of a confidence threshold, ASMOV^x refers to ASMOV with the semantic verification component turned off.

6 Directory

The directory test case aims at providing a challenging task for ontology matchers in the domain of large directories to show whether ontology matching tools can effectively be applied for the integration of "shallow ontologies". The focus of this task is to evaluate performance of existing matching tools in real world taxonomy integration scenario.

6.1 Test set

As in previous years [9; 11; 4], the data set exploited in the directory matching task was constructed from Google, Yahoo and Looksmart web directories following the methodology described in [13]. The data set is presented as taxonomies where the nodes of the web directories are modeled as classes and classification relation connecting the nodes is modeled as an `rdfs:subClassOf` relation.

The key idea of the data set construction methodology is to significantly reduce the search space for human annotators. Instead of considering the full matching task which is very large (Google and Yahoo directories have up to $3 * 10^5$ nodes each: this means that the human annotators need to consider up to $(3*10^5)^2 = 9*10^{10}$ correspondences), it uses semi automatic pruning techniques in order to significantly reduce the search space. For example, for the data set described in [13], human annotators consider only 2265 correspondences instead of the full matching problem.

The specific characteristics of the data set are:

- More than 4.500 node matching tasks, where each node matching task is composed from the paths to root of the nodes in the web directories.
- Reference alignment for all the matching tasks.
- Simple relationships, in particular, web directories contain only one type of relationships, which is the so-called classification relation.
- Vague terminology and modeling principles, thus, the matching tasks incorporate the typical real world modeling and terminological errors.

6.2 Results

In OAEI 2009, 7 out of 16 matching systems participated on the web directories test case, while in OAEI-2008, 7 out of 13, in OAEI 2007, 9 out of 18, in OAEI 2006, 7 out of 10, and in OAEI 2005, 7 out of 7 did it.

Precision, recall and F-measure results of the systems are shown in Figure 4. These indicators have been computed following the TaxMe2 [13] methodology, with the help of the Alignment API [8], version 3.4.

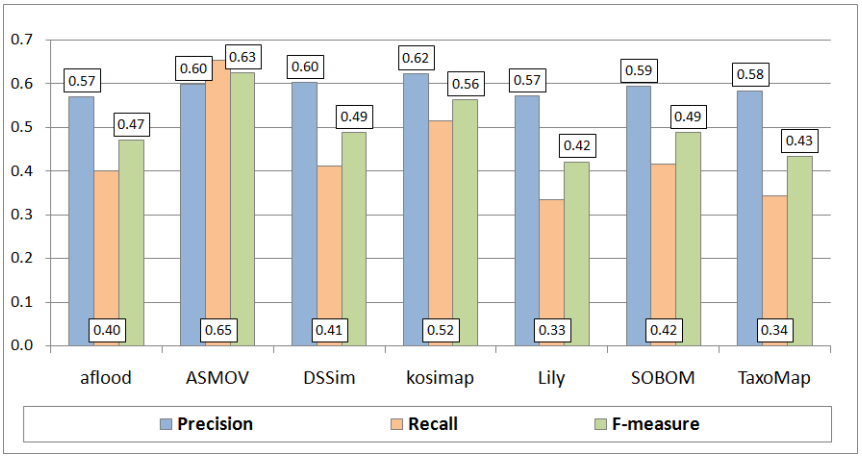


Fig. 4. Matching quality results.

We can observe from Table 16, that in general the systems that participated in the directory track in 2008 (DSSim, Lily and TaxoMap), have either maintained or decreased their precision and recall values. The only system that increased its recall value is ASMOV. In fact, ASMOV is the system with the highest F-measure value in 2009.

Table 16 shows that in total 24 matching systems have participated in the directory track during the 5 years (2005 – 2009) of the OAEI campaigns. No single system has participated in all campaigns involving the web directory dataset (2005 – 2009). A total of 16 systems have participated only one time in the evaluation, only 3 systems have participated 2 times, and 5 systems have participated 3 times.

As can be seen in Figure 5 and Table 16, there is an increase in the average precision for the directory track up to 2008, remaining constant in 2009. The average recall in 2009 increased in comparison to 2008, but the highest average recall remains that of 2007. Considering F-measure, results for 2009 show the highest average in the 4 years (2006 to 2009). Notice that in 2005 the data set allowed only the estimation of recall, therefore Figure 5 and Table 16 do not contain values of precision and F-measure for 2005.

A comparison of the results in 2006, 2007, 2008 and 2009 for the top-3 systems of each year based on the highest values of the F-measure indicator is shown in Figure 6. The key observation here is that even though two of the top-3 systems of 2008 (Lily and DSSim) participated in the directory task this year, they did not manage to get into the top-3, indicating an overall increase of performance by the total set of participating

System	Recall					Precision				F-Measure			
Year →	2005	2006	2007	2008	2009	2006	2007	2008	2009	2006	2007	2008	2009
aflood					0.40				0.57				0.47
ASMOV			0.44	0.12	0.65		0.59	0.64	0.60		0.50	0.20	0.63
automs		0.15				0.31				0.20			
CIDER				0.38				0.60				0.47	
CMS	0.14												
COMA		0.27				0.31				0.29			
ctxMatch2	0.09												
DSSim			0.31	0.41	0.41		0.60	0.60	0.60		0.41	0.49	0.49
Dublin20	0.27												
Falcon	0.31	0.45	0.61			0.41	0.55			0.43	0.58		
FOAM	0.12												
HMatch		0.13				0.32				0.19			
kosimap					0.52				0.62				0.56
Lily			0.54	0.37	0.33		0.57	0.59	0.57		0.55	0.46	0.42
MapPSO				0.31				0.57				0.40	
OCM		0.16				0.33				0.21			
OLA	0.32		0.84				0.62				0.71		
OMAP	0.31												
OntoDNA			0.03				0.55				0.05		
Prior		0.24	0.71			0.34	0.56			0.28	0.63		
RiMOM		0.40	0.71	0.17		0.39	0.44	0.55		0.40	0.55	0.26	
SOBOM					0.42				0.59				0.49
TaxoMap				0.34	0.34			0.59	0.59			0.43	0.43
X-SOM			0.29				0.62				0.39		
Average	0.22	0.26	0.50	0.30	0.44	0.35	0.57	0.59	0.59	0.29	0.49	0.39	0.50
#	7	7	9	7	7	7	9	7	7	7	9	7	7

Table 16. Summary of submissions by year (no precision was computed in 2005). The Prior line covers Prior+ as well and the OLA line covers OLA_2 as well.

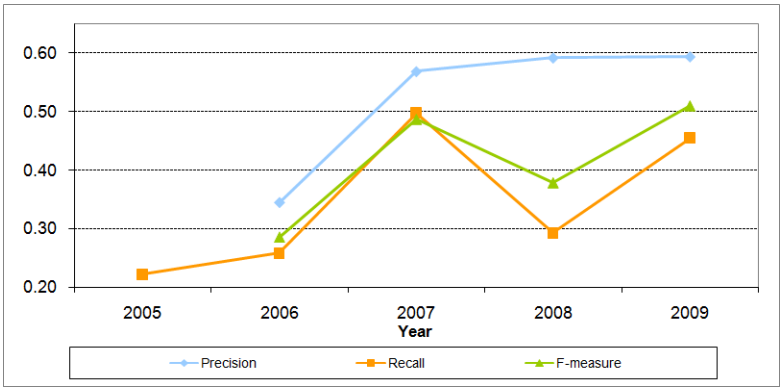


Fig. 5. Average results of the top-3 systems per year.

systems this year. As can be seen in Table 16, DSSim maintained its performance of 2008, having the same F-measure as SOBOM (a newcomer and 3rd place of 2009), only 1% less of recall than SOBOM, but 1% more of precision. ASMOV increased its F-measure, presenting the highest value for this year directory track, and in overall in its 3 years of participation. The second place corresponds to kosimap, also a newcomer.

The quality of the best F-measure result of 2009 (0.63) achieved by ASMOV is higher than the best F-measure of 2008 (0.49) demonstrated by DSSim and higher than that of 2006 by Falcon (0.43), but still lower than the best F-measure of 2007 (0.71) by OLA_2. The best precision result of 2009 (0.62) achieved by kosimap is lower than the best precision value of 2008 (0.64) demonstrated by ASMOV and equal to the results obtained in 2007 by both OLA_2 and X-SOM. Finally, for what concerns recall, the best result of 2009 (0.65) achieved by ASMOV is higher than the best value of 2008 (0.41) demonstrated by DSSim and the best value in 2006 (0.45) by Falcon, but still lower than the best result obtained in 2007 (0.84) obtained by OLA_2.

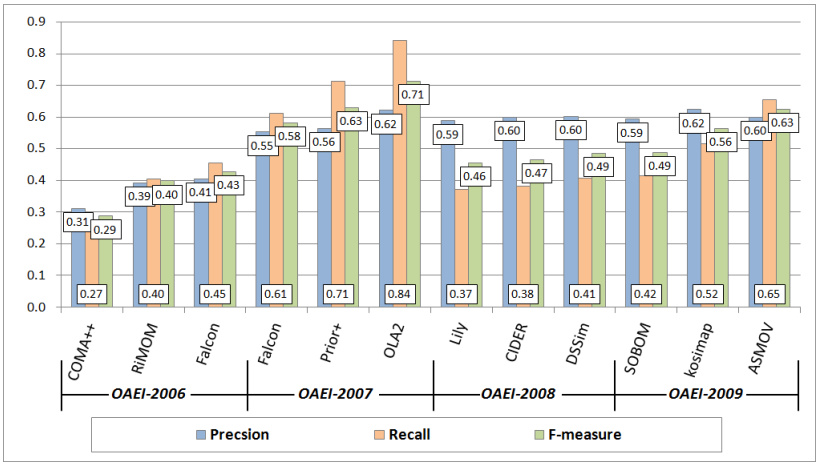


Fig. 6. Comparison of matching quality results in 2006, 2007, 2008 and 2009.

Partitions of positive and negative correspondences according to the system results are presented in Figures 7 and 8, respectively.

Figure 7 shows that the systems managed to discover only 68% of the total number of positive correspondences (Nobody = 32%). Only 26% of positive correspondences were found by all seven participating systems. The percentage of positive correspondences found by the systems this year is higher than the values of 2008, when 54% of the positive correspondences where found. Figure 8 shows that more than half (56%) of the negative correspondences were not found by the systems (correctly) in comparison to 66% not found in 2008. Figure 8 also shows that all participating systems found 17% of the negative correspondences, i.e., mistakenly returned them as positive. The last two observations suggest that the discrimination ability of the dataset remains still high as in previous years.

Let us now compare partitions of the system results in 2006, 2007, 2008 and 2009 on positive and negative correspondences, see Figures 9 and 10, respectively. Figure 9

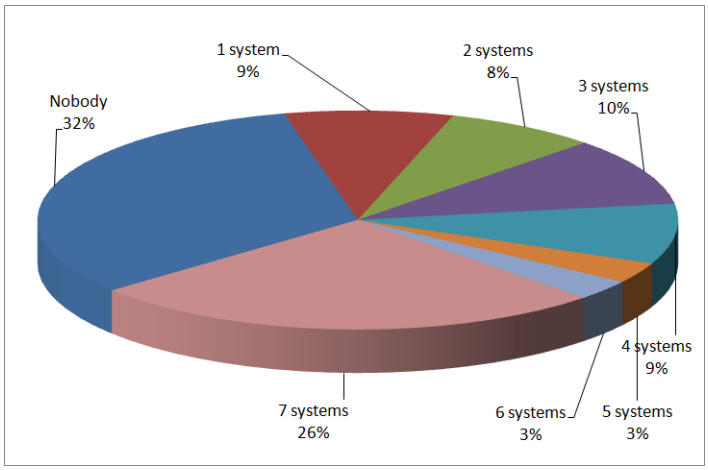


Fig. 7. Partition of the system results on positive correspondences.

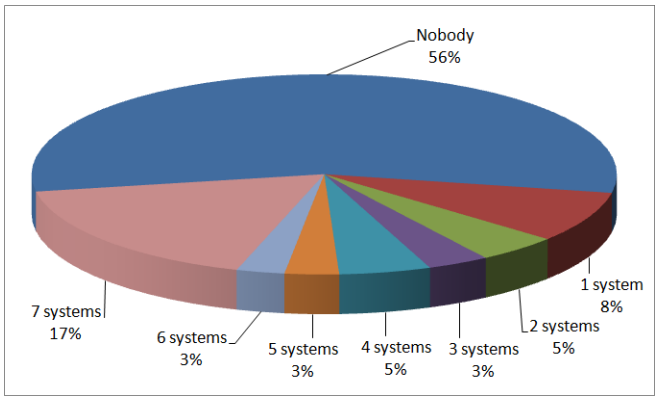


Fig. 8. Partition of the system results on negative correspondences.

shows that 32% of positive correspondences have not been found by any of the matching systems this year. This value is better that the values of 2006 (43%) and 2008 (46%). In 2007 all the positive correspondences have been collectively found; these results (2007) were exceptional because the participating systems all together had a full coverage of the expected results and very high precision and recall. Unfortunately, the best systems of 2007 did not participate this year (nor in 2008) and the other systems do not seem to cope with the results of 2007.

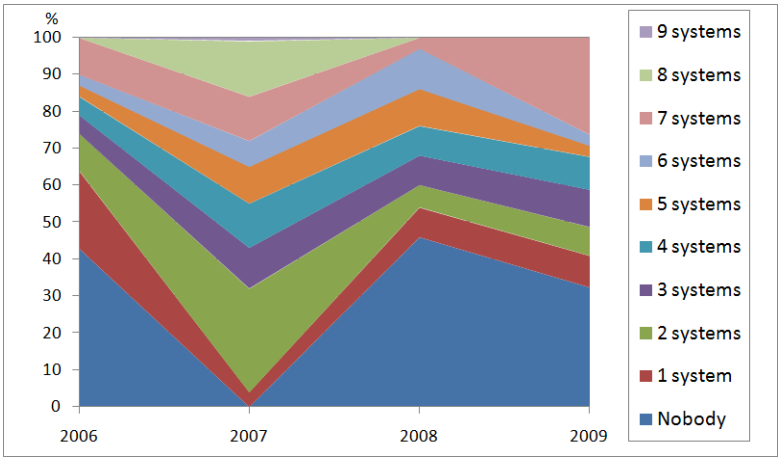


Fig. 9. Comparison of partitions of the system results on positive correspondences in 2006, 2007, 2008 and 2009.

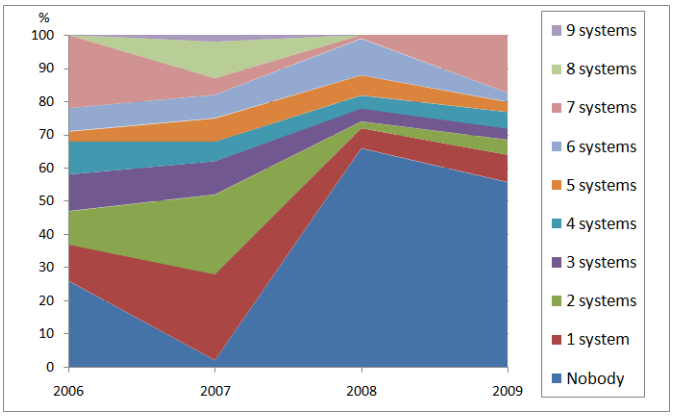


Fig. 10. Comparison of partitions of the system results on negative correspondences in 2006, 2007, 2008 and 2009.

Figure 10 shows that this year 56% of the negatives correspondences were correctly not found. There is a decrease in comparison to the value of 2008, when 66% of the negatives correspondences where not found, being the best value in all years (2006 to 2009). This year 17% of the negative correspondences were mistakenly found by all

the (7) participating systems, being the best value that of last year (1%). An interpretation of these observations could be that the set of participating systems in 2009 have a more cautious strategy than in 2007 and 2006, but still a little bit more brave than in 2008. In 2007, we can observe that the set systems showed the most brave strategy in discovering correspondences of all the yearly evaluation initiatives, when the set of positive correspondences was fully covered, but covering mistakenly also 98% of the negative correspondences. This year the behavior of the overall systems is more similar (but better) to the behavior of the overall set of participating systems in 2008.

6.3 Comments

This year the average performance of the systems (given by F-measure in Figure 5) is the best of all 4 years (2006 to 2009). This suggests that the set of participating systems have found a balance between a brave and cautious behavior for discovering correspondences. However, the value for the F-measure (0.51) indicates that there is still room for further improvements. Finally, as partitions of positive and negative correspondences indicate (see Figure 7 and Figure 8), the dataset still retains a good discrimination ability, i.e., different sets of correspondences are still hard for the different systems.

7 Library

This task, organized in the context of the TELplus[12] project, focuses on a case for which the MACS[13] project established a (partial) manual reference alignment. Participants of this task had to create pairwise alignments between three large subject heading lists in different languages. The required alignments links were SKOS relations. This task is similar, from a methodological perspective, to the OAEI 2008 Library track. It uses however a different dataset.

7.1 Test data

The vocabularies to match are:

- LCSH, the Library of Congress Subject Headings, available as linked data at `http://id.loc.gov`. Contains around 340K concepts, including 250K general subjects.
- RAMEAU, the heading list used at the French National Library, available as linked data at `http://stitch.cs.vu.nl/rameau`. Contains around 150K concepts, including 90K general subjects.
- SWD, the heading list used at the German National Library. Contains 800K concepts, including 160K general subjects.

[12] http://www.theeuropeanlibrary.org/telplus
[13] http://macs.cenl.org

The concepts from the three vocabularies are used as subjects of books. For each concept, the usual SKOS lexical and semantic information is provided: preferred labels, synonyms and notes, broader and related concepts, etc. The three subject heading lists have been represented according to the SKOS model, but an OWL version has also been made available. Note that even though two of these vocabularies are available online as RDF data, we have provided dumps for the convenience of participants.

We have also made available a part of the MACS manual correspondences between these vocabularies, which can be used as a *learning set*. However, none of the participants asked for it.

7.2 Evaluation and results

Only one team handed in final results: TaxoMap, which produced results as listed in Table 17.

Type of relation	LCSH-RAMEAU	RAMEAU-SWD	LCSH-SWD
exactMatch	5,074	1,265	38
broadMatch	116,789	17,220	0
narrowMatch	48,817	6,690	0
relatedMatch	13,205	1,317	0

Table 17. Taxomap results.

We have followed the dual evaluation approach of the previous 2008 Library Track, which featured a "thesaurus merging" evaluation (based on a post-hoc partial reference alignment) and a "re-indexing" one (assessing the use of correspondences for translating subject annotations from one thesaurus to another). The main difference is that the first evaluation method has now been replaced by comparing to an already existing partial reference alignment (the MACS one), avoiding to manually assess the participant's results.

Comparing with partial reference alignment (MACS) As no participant used the training set we provided, we use the complete MACS correspondences as reference alignment. In the version we received (MACS is still currently adding manual correspondences to this reference set), this reference alignment comprised 87,183 LCSH-RAMEAU correspondences, 13,723 RAMEAU-SWD correspondences, and 12,203 LCSH-SWD correspondences.

Table 18 shows the results when taking into account all correspondences that belong to a certain relation selection. For a given relation selection, the token "–" means that no extra relation was provided at that level, hence the results are identical to the ones of the previous selection level. Cov. refers to the coverage, that is, the percentage of MACS correspondences which were found in the evaluated alignment.

Table 19 shows the results obtained when selecting only the "best" available correspondences for one concept (that is, the one with the highest confidence measure), and discarding the others.

TaxoMap links evaluated	LCSH-RAMEAU		RAMEAU-SWD		LCSH-SWD	
	Prec.	Cov.	Prec.	Cov.	Prec.	Cov.
exactMatch	72.1	5.7	27.1	1.4	44.4	0.03
eM + broadMatch	3.6	6.9	2.3	1.9	–	–
eM + bM + narrowMatch	2.8	7.3	1.8	2.0	–	–
all relations	2.7	7.5	1.9	2.2	–	–

Table 18. Results for comparison with MACS (percentage) – using all correspondences.

TaxoMap links evaluated	LCSH-RAMEAU		RAMEAU-SWD		LCSH-SWD	
	Prec.	Cov.	Prec.	Cov.	Prec.	Cov.
exactMatch	78.7	5.7	39.5	1.4	44.4	0.03
eM + broadMatch	22.0	6.0	13.5	1.6	–	–
eM + bM + narrowMatch	14.4	5.9	10.8	1.6	–	–
all relations	13.4	5.8	10.9	1.7	–	–

Table 19. Results for comparison with MACS (percentage) – using only the best correspondences for each concept.

Results for the re-indexing scenario The second usage scenario is based on an *annotation translation* process supporting the re-indexing of books indexed with one vocabulary, using concepts from the mapped vocabulary (see [14]). Here we use book annotations from the British Library (using LCSH), the French National Library (using RAMEAU) and the German National Library (using SWD), see Table 19(a).

For each pair of vocabularies A-B, this scenario interprets the correspondences as rules to translate existing book annotations with A into equivalent annotations with B. In the case at hand, the book collections have a few books in common (cf. Table 19(b)), which are therefore described according to two vocabularies. Based on the quality of the results for those books for which we know the correct annotations, we can assess the quality of the initial correspondences.

(a) Collections and books with subject annotations.

Collection	Books with subject annotation
English	2,448,050
French	1,457,143
German	1,364,287

(b) Common books between different collections.

Collection pair	Common books
French–English	182,460
German–English	83,786
German–French	63,340

Table 20. Data on collections.

Evaluation settings and measures. For each pair of vocabularies A-B, the simple concept-to-concept correspondences sent by participants were transformed into more complex mapping rules that associate one concept from A with a set of concepts from B – as some concepts are involved in several correspondences.

The set of A concepts attached to each book is then used to decide whether these rules are *fired* for this book. If the A concept of one rule is contained by the A annotation of a book, then the rule is fired. As several rules can be fired for a same book, the union of the consequents of these rules forms the translated B annotation of the book.

On a set of books selected for evaluation, the generated concepts for a book are then compared to the ones that are deemed correct for this book. At the annotation level, we measure the precision, the recall, and the Jaccard overlap measure (Jac.) between the produced annotation and the correct one.

In the formulas used, results are counted on a book and annotation basis, and not on a rule basis. This reflects the importance of different thesaurus concepts: a translation rule for a frequently used concept is more important than a rule for a rarely used concept.

Results. Table 21 shows the results when taking into account all correspondences that belong to a certain relation selection.

TaxoMap links evaluated	LCSH-RAMEAU			RAMEAU-SWD			LCSH-SWD		
	Prec.	Rec.	Jac.	Prec.	Rec.	Jac.	Prec.	Rec.	Jac.
exactMatch	22.3	6.1	5.5	14.2	3.1	2.4	1.3	0.003	0.002
eM + broadMatch	2.1	7.8	1.5	2.3	3.6	1.1	–	–	–
eM + bM + narrowMatch	1.2	9.2	1.0	0.8	3.9	0.5	–	–	–
all relations	1.1	9.3	0.9	0.7	4.0	0.5	–	–	–

Table 21. Re-indexing evaluation results (percentage) – using all correspondences.

Table 22 shows the results obtained when selecting only the "best" available mapping for one concept and discarding the others.

TaxoMap links evaluated	LCSH-RAMEAU			RAMEAU-SWD			LCSH-SWD		
	Prec.	Rec.	Jac.	Prec.	Rec.	Jac.	Prec.	Rec.	Jac.
exactMatch	22.8	5.8	5.3	14.2	1.9	1.7	1.2	0.002	0.002
eM + broadMatch	10.2	6.0	4.9	6.9	2.0	1.7	–	–	–
eM + bM + narrowMatch	7.2	4.5	3.3	5.9	1.9	1.5	–	–	–
all relations	6.4	4.0	2.9	5.8	1.9	1.5	–	–	–

Table 22. Re-indexing evaluation results (percentage) – using all Taxomap correspondences.

7.3 Discussion

The setting for this year's library task clearly shows the limits of current matching tools. The case at hand, mostly because of its size and its multilingual aspect, is extremely difficult to handle. The performance of TaxoMap, from this perspective, should be regarded as a significant achievement, as it was the only one to manage to ingest hundreds of concepts and return alignments between them.

The results of TaxoMap, which could not apply its usual partition approach, and uses to a great extent automatic translation, are not very good. More precisely, they

are especially weak when relations other than strict equivalence are considered, highlighting the value of being able to sort mapping results using the type of relation or the strength of the confidence measure granted to correspondences–options which are both offered by TaxoMap. Both precision and coverage/recall are low for the non-equivalence correspondences, even though they bring a huge number of potential matches. The translation could give better results for the equivalent correspondences, at the cost of coverage of course.

It is worth mentioning that as last year, the results for the comparison with a reference mapping and the re-indexing evaluation largely differ, showing that correspondences have a different relevance depending on the application scenario. correspondences based on translation will perform obviously better for scenarios where the intension of concepts matters, rather than for cases where their actual usage in book collections should be carefully taken into account.

8 Oriented alignment

This year we introduced evaluation of alignments containing other relations that the classical equivalence between entities, e.g., subsumption relations.

8.1 Test data

The first dataset (dataset 1) has been derived from the benchmark series of the OAEI 2006 campaign [9] and was created for the evaluation of the "Classification-Based Learning of Subsumption Relations" (CSR) method. As a configuration of CSR exploits the properties of concepts (for the cases where properties are used as features), we do not include the OAEI 2006 ontologies whose concepts have no properties. Furthermore, we have excluded from the dataset the OAEI ontologies with no defined subsumption relations among their concepts. This is done because CSR exploits the subsumption relations in the input ontologies to generate training examples. More specifically, all benchmarks (101-304) except 301 to 304, define the second ontology of each pair as an alteration of the same ontology, i.e., the first one, numbered 101.

The second dataset (dataset 2) is composed of 45 pairs of real-world ontologies coming from the Consensus Workshop track of the OAEI 2006 campaign (all pairwaise combinations). The domain of the ontologies concerns the organization of conferences and they have been developed within the OntoFarm project[7].

The reference alignment for all datasets has been manually created by knowledge engineers. The major guidelines that were followed for the location of subsumption relations are as follows: (a) use existing equivalences in order to find inferred subsumptions, and (b) understand the "intended meaning" of the concepts, e.g., by inspecting specifications and relevant information attached to them. The format of the reference alignment is the Alignment format as used in the benchmark series.

8.2 Participants

Three systems returned results for the first dataset, namely, ASMOV, RiMoM and TaxoMap. We present these results by also presenting the results achieved by CSR (as a comparison basis), presenting also the results of CSR for the second dataset.

8.3 Results

system	CSR			ASMOV			RiMoM			TaxoMap		
test	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.
1xx	0.97	0.97	0.97	1.00	1.00	1.00	1.00	1.00	1.00	NaN	0	NaN
2xx	0.84	0.78	0.80	0.94	0.94	0.94	0.67	0.85	0.69	0.84	0.08	0.25
3xx	0.66	0.72	0.69	0.86	0.60	0.60	0.59	0.81	0.64	0.72	0.11	0.17
Average	0.83	0.79	0.80	0.94	0.90	0.93	0.69	0.86	0.71	0.63	0.07	0.23

Table 23. Results of all systems when applied to data set 1.

Table 23 presents the precision, recall and F-measure values, of each participating system in all tests (average) and separately in each test category, e.g., 1xx. We observe that in terms of F-measure ASMOV achieves the best results, followed by CSR, RiMoM and then by TaxoMap. Also, we observe that although CSR has a higher precision than RiMoM, RiMoM has a higher recall. ASMOV and RiMoM did not make specific changes to their methods for this dataset. TaxoMap exploits the lexicalizations of concepts to compute subsumption relations. Furthermore, CSR does not exploit equivalence relations.

Concerning dataset 2, Table 24 depicts the precision and recall values for each pair of ontologies in the dataset provided by CSR. The other methods did not provide results for this dataset. An observation is that the performance of CSR is worst in this dataset, in comparison to the first dataset.

9 Instance matching

For the first time in OAEI, an instance matching track was proposed to participants. The aim of this track is to evaluate matchers on instance data coming from diverse sources. Both data extracted from published Web datasets, and a testbed presenting various automatically generated values and structure modifications were proposed.

9.1 AKT-Rexa-DBLP

The AKT-Rexa-DBLP (ARS) test case aims at testing the capability of the tools to match individuals. All three datasets were structured using the same schema. The challenges for the matchers included ambiguous labels (person names and paper titles) and noisy data (some sources contained incorrect information).

Ontology pair	Prec.	Rec.	Ontology pair	Prec.	Rec.
Iasted-Cmt	0.6	0.7	Confious-Sigkdd	0.26	0.51
Cmt-confOf	0.76	0.83	crs_dr-Sigkdd	0.09	0.13
Cmt-Confious	0.28	0.31	Iasted-Sigkdd	0.17	0.88
confOf-Confious	0.14	0.47	OpenConf-Sigkdd	0.22	0.39
crs_dr-Confious	0.08	0.11	Pcs-Sigkdd	0.18	0.48
Iasted-Confious	0.08	0.25	Cmt-Conference	0.25	0.11
OpenConf-Confious	0.22	0.45	confOf-Conference	0.43	0.29
Pcs-Confious	0.16	0.43	Confious-Conference	0.15	0.43
Cmt-crs_dr	0.54	0.39	crs_dr-Conference	0.58	0.11
confOf-crs_dr	0.38	0.38	Iasted-Conference	0.2	0.08
confOf-Iasted	0.47	0.38	OpenConf-Conference	0.14	0.15
crs_dr-Iasted	0.18	0.38	Pcs-Conference	0.05	0.05
OpenConf-Iasted	0.15	0.38	Sigkdd-Conference	0.15	0.19
Pcs-Iasted	0.21	0.39	Cmt-ekaw	0.46	0.72
Cmt-OpenConf	0.32	0.41	confOf-ekaw	0.51	0.74
confOf-OpenConf	0.22	0.39	Confious-ekaw	0.22	0.59
crs_dr-OpenConf	0.15	0.32	crs_dr-ekaw	0.21	0.2
Cmt-Pcs	0.47	0.77	Iasted-ekaw	0.32	0.33
confOf-Pcs	0.24	0.47	OpenConf-ekaw	0.28	0.28
crs_dr-Pcs	0.17	0.69	Pcs-ekaw	0.36	0.67
OpenConf-Pcs	0.1	0.26	Sigkdd-ekaw	0.64	0.78
Cmt-Sigkdd	0.54	0.81	Conference-ekaw	0.58	0.65
confOf-Sigkdd	0.29	0.64			
Average				0.29	0.43

Table 24. Results of CSR when applied to dataset 2.

Test set The test case included three datasets from the domain of scientific publications:

- AKT EPrints archive[14]. This dataset contains information about papers produced within the AKT research project.
- Rexa dataset[15]. This dataset was extracted from the Rexa search server, which was constructed at the University of Massachusetts using automatic information extraction algorithms.
- SWETO DBLP dataset[16]. This is a publicly available dataset listing publications from the computer science domain.

The SWETO-DBLP dataset was originally represented in RDF. Two other datasets (AKT EPrints and Rexa) were extracted from the HTML sources using specially constructed wrappers and structured according to the SWETO-DBLP ontology[17]. The ontology describes information about scientific publications and their authors and extends the commonly used FOAF ontology[18]. Authors are represented as individuals of the

[14] http://eprints.aktors.org/
[15] http://www.rexa.info/
[16] http://lsdis.cs.uga.edu/projects/semdis/swetodblp/
[17] http://lsdis.cs.uga.edu/projects/semdis/swetodblp/august2007/opus_august2007.rdf
[18] http://xmlns.com/foaf/spec/

foaf:Person class, and a special class *sweto:Publication* is defined for publications, with two subclasses *sweto:Article* and *sweto:Article_in_Proceedings* for journal and conference publications respectively. The participants were invited to produce alignments for each pair of datasets (AKT/Rexa, AKT/DBLP, and Rexa/DBLP).

Evaluation results Five participants submitted results for the AKT-Rexa-DBLP test case produced by their systems: DSSim, RiMOM, FBEM, HMatch, and ASMOV. The results were evaluated by comparing them with a manually constructed reference alignment and calculating the standard precision, recall, and F-measure. We measured the performance of each system for the classes *sweto:Publication* and *foaf:Person* separately, as well as for the combined set of individuals. These evaluation results are provided in Table 25.

	sweto:Publication			foaf:Person			Overall		
System	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.
AKT/Rexa									
DSSim	0.15	0.16	0.16	0.81	0.30	0.43	0.60	0.28	0.38
RiMOM	1.00	0.72	0.84	0.92	0.70	0.79	0.93	0.70	0.80
FBEM	0.99	0.61	0.76	0.73	0.02	0.03	0.94	0.10	0.18
HMatch	0.97	0.89	0.93	0.94	0.39	0.56	0.95	0.46	0.62
ASMOV	0.32	0.79	0.46	0.76	0.24	0.37	0.52	0.32	0.39
AKT/DBLP									
DSSim	0	0	0	0.15	0.19	0.17	0.11	0.15	0.13
RiMOM	0.96	0.97	0.96	0.93	0.50	0.65	0.94	0.59	0.73
FBEM	0.98	0.80	0.88	0	0	0	0.98	0.16	0.28
HMatch	0.93	0.97	0.95	0.58	0.57	0.57	0.65	0.65	0.65
Rexa/DBLP									
DSSim	0	0	0	0	0	0	0	0	0
RiMOM	0.94	0.95	0.94	0.76	0.66	0.71	0.80	0.72	0.76
FBEM	0.98	0.15	0.26	1.00	0.11	0.20	0.99	0.12	0.21
HMatch	0.45	0.96	0.61	0.40	0.34	0.37	0.42	0.48	0.45

Table 25. Results of AKT-Rexa-DBLP test case.

The AKT/Rexa test scenario was the only one for which the results for ASMOV were available and the only one for which all the systems provided alignments for both *foaf:Person* and *sweto:Publication* classes. FBEM for the AKT/DBLP test case only produced alignments for *Publication* instances, which reduced their overall recall. For the class *Publication* the best F-measure in all three cases was achieved by RiMOM with HMatch being the second. FBEM, which specifically focused on precision, achieved the highest precision in all three cases at the expense of recall. It is interesting to see the difference between systems in the Rexa/DBLP scenario where many distinct individuals had identical titles, e.g., "Editorial.", or "Minitrack Introduction.". This primarily affected the precision in the case of HMatch and RiMOM, but reduced recall for FBEM.

The performance of all systems was lower for the class *Person* where ambiguous personal names and different label formats reduced the performance of string similarity

techniques. The highest F-measure was achieved by RiMOM and by HMatch for the three test cases. Again, it is interesting to note the difference between RiMOM, HMatch, and FBEM in the Rexa/DBLP case where the first two systems focused on F-measure and the second one on precision. This distinction of approaches can be an important criterion when a tool has to be selected for a real world use case: in some cases the cost of an erroneous correspondence is much higher than than the cost of a missed one, e.g., the large-scale entity naming service such as FBEM, while in other scenarios this might not be true, e.g., assisting the user who performs manual alignment of datasets. In contrast, in the AKT/Rexa scenario the performance of FBEM was lower than the performance of other systems both in terms of precision and recall. This was caused by different label formats used by AKT and Rexa datasets ("FirstName LastName" vs "LastName, FirstName"), which affected FBEM.

Because in all three scenarios the datasets had more *Person* individuals than *Publication* ones, the overall results were primarily influenced by the performance of the tools on the class *Person*. Again, HMatch and RiMOM had the highest F-measure for all the test cases. We can see a comparison with respect to F-measure in Figure 11.

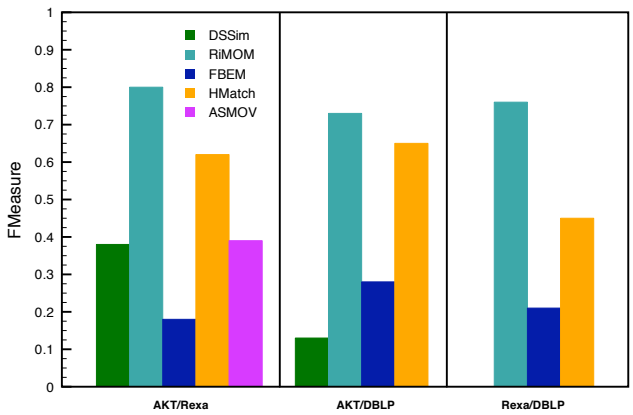


Fig. 11. Comparison on AKT-Rexa-DBLP with respect of FMeasure

9.2 ISLab Instance Matching Benchmark

The ISLab Instance Matching Benchmark (IIMB) is a benchmark automatically generated starting from one data source that is automatically modified according to various criteria. The original data source contains OWL/RDF data about actors, sport persons, and business firms provided by the OKKAM European project[19]. The benchmark is composed by 37 test cases. For each test case we require participants to match the original data source against a new data source. The original data source contains about 200 different instances. Each test case contains a modified version of the original data source and the corresponding reference alignment containing the expected results. Modifications introduced in IIMB are the following:

[19] http://www.okkam.org

– Test case 001: Contains an identical copy of the original data source (instance IDs are randomly changed).
– Test case 002 - Test case 010: Value transformations, i.e., typographical errors simulation, use of different standard for representing the same information. In order to simulate typographical errors, property values of each instance are randomly modified. Modifications are applied on different subsets of the instances property values and with different levels of difficulty, i.e., introducing a different number of errors.
– Test case 011 - Test case 019: Structural transformations, i.e., deletion of one or more values, transformation of datatype properties into object properties, separation of a single property into more properties.
– Test case 020 - Test case 029: Logical transformations, i.e., instantiation of identical individuals into different subclasses of the same class, instantiation of identical individuals into disjoint classes, instantiation of identical individuals into different classes of an explicitly declared class hierarchy.
– Test case 030 - Test case 037: Several combinations of the previous transformations.

Evaluation results. In this first edition of the instance matching track, six systems participated in the IIMB task, namely AFlood, ASMOV, DSSim, HMatch, FBEM, and RiMOM. In Table 26, we provide real precision and recall measures for the participating systems.

System	AFlood			ASMOV			DSSim		
Test	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.
002 - 010	1.00	0.99	0.99	1.00	1.00	1.00	1.00	0.37	0.54
011 - 019	0.90	0.72	0.80	0.99	0.92	0.96	0.99	0.28	0.43
020 - 029	0.85	1.00	0.92	1.00	1.00	1.00	0.85	0.99	0.91
030 - 037	0.94	0.75	0.83	1.00	0.98	0.99	1.00	0.30	0.46
H-means	0.92	0.87	0.89	1.00	0.98	0.99	0.92	0.48	0.63

System	HMatch			FBEM			RiMOM		
Test	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.
002 - 010	0.97	0.98	0.97	0.95	0.93	0.94	1.00	1.00	1.00
011 - 019	0.88	0.83	0.85	0.78	0.52	0.62	1.00	0.93	0.97
020 - 029	0.78	1.00	0.88	0.08	1.00	0.15	0.85	1.00	0.92
030 - 037	0.94	0.89	0.92	0.10	0.53	0.16	1.00	0.99	0.99
H-means	0.89	0.93	0.91	0.16	0.75	0.27	0.96	0.98	0.97

Table 26. IIMB results: precision and recall.

A first general remark about the results is that three of the participating systems, i.e., AFlood, ASMOV, and DSSim, provide better results in terms of precision rather than in terms of recall, even if AFlood and ASMOV results can be considered very good in both. On the other end, HMatch, FBEM, and RiMOM provide better results in terms of recall, with better performances in case of HMatch and RiMOM. Coming to the four categories of test cases, we can conclude that all the six systems show very good performances on cases 002 - 010, where we just introduced some data errors by

maintaining both the data structure and the logical properties of data. On test cases 011 - 019, where data structures were changed by deleting or modifying property assertions, AFlood, ASMOV, HMatch, and RiMOM still perform over 80% in terms of F-Measure, while both DSSim and FBEM performances are lower, especially with respect to recall. In general, test cases 011 - 019 were more difficult with respect to recall than to precision. Test cases 020 - 029 were focused on logical transformations. In order to achieve good performances here, it is important to take into account logical implications of the schema over the instances. This is achieved by AFlood, ASMOV, DSSim, and RiMOM. HMatch maintains high recall and good precision, while FBEM's precision seems very low. Finally, test cases 030 - 037 as well as the final harmonic mean shown, AFlood, ASMOV, HMatch, and RiMOM provide good results both in terms of precision and in terms of recall. DSSim is more effective on precision, while FBEM is stronger in terms of recall.

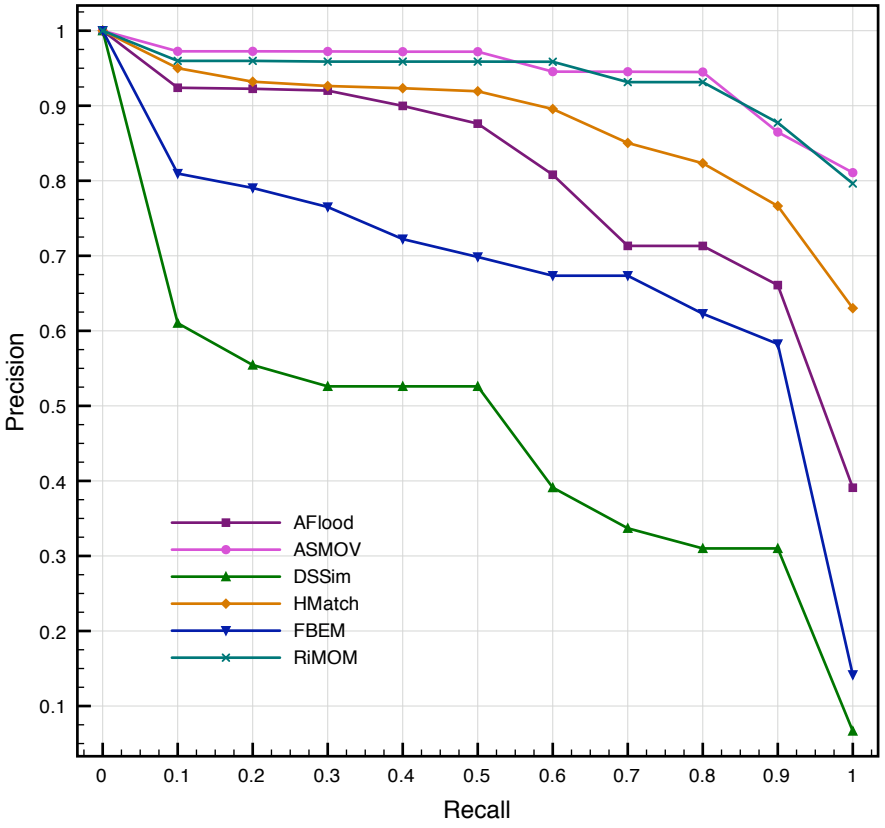


Fig. 12. Precision/recall graphs. They cut the results given by the participants under a threshold necessary for achieving $n\%$ recall and compute the corresponding precision.

All the six systems provided their results with confidence measures. It is thus possible to draw precision/recall graphs in order to compare them (see Figure 12). The graph is computed by averaging the graphs of each of the tests. The precision/recall graph confirms the comparison done over real precision and recall values, especially in case of recall values lower than 50%. After that threshold, ASMOV, RiMOM, and HMatch maintain their performances high, and FBEM performances are stable. Instead, DSSim and AFlood values of precision decrease quite quickly, even if AFlood performances are still better than FBEM and DSSim.

10 Very Large Crosslingual Resources

The goal of the Very Large Crosslingual Resources challenge is twofold. First, we are interested in matching vocabularies in different languages. Many collections throughout Europe are indexed with vocabularies in languages other than English. These collections would benefit from an alignment to resources in other languages to broaden the user group, and possibly enable integrated access to the different collections. Second, we intend to present a realistic use case in the sense that the resources are large, rich in semantics but weak in formal structure, i.e. realistic on the Web. For collections indexed with an in-house vocabulary, the link to a widely-used and rich resource can enhance the structure and increase the scope of the in-house thesaurus. In this task, we aim for skos:exactMatch and skos:closeMatch relations.

10.1 Test data

Three resources are used in this task:

WordNet WordNet is a lexical database of the English language developed at Princeton University[20]. Its main building blocks are synsets: groups of words with a synonymous meaning. In this task, the goal is to match noun-synsets. WordNet contains 7 types of relations between noun-synsets, but the main hierarchy in WordNet is built on hyponym relations, which are similar to subclass relations. W3C has translated WordNet version 2.0 into RDF/OWL.

The original WordNet model is a rich and well-designed model. However, some tools may have problems with the fact that the synsets are instances rather than classes. Therefore, for the purpose of this OAEI task, we have translated the hyponym hierarchy in a skos:broader hierarchy, making the synsets skos:Concepts.

DBpedia DBpedia contains 2.18 million resources or "things", each tied to an article in the English language Wikipedia. The "things" are described by titles and abstracts in English and often also in Dutch. DBpedia "things" have numerous properties, such as categories, properties derived from the wikipedia "infoboxes", links between pages within and outside wikipedia, etc.

GTAA The GTAA is a Dutch thesaurus used by the Netherlands Institute for Sound and Vision to index their collection of TV programs. It is a facetted thesaurus, of which we use the following four facets: (1) **Subject**: the topic of a TV program,

[20] `http://wordnet.princeton.edu/`

≈3800 terms; (2) **People**: the main people mentioned in a TV program, ≈97.000 terms; **Names**: the main "Named Entities" mentioned in a TV program (Corporation names, music bands, etc.), ≈27.000 terms; **Location**: the main locations mentioned in a TV program or the place where it has been created, ≈14.000 terms.

The purpose of this task is to match GTAA concepts to DBpedia "things" and WordNet synsets.

10.2 Evaluation setup

We evaluate the results of the two alignments (GTAA-WordNet, GTAA-DBpedia) in terms of precision and recall. Aside from an overall measure, we also present measures for each GTAA facet separately. We introduce an evaluation on a 3-point scale of 0 - 0.5 - 1. We assign 1 point when the relation between two concepts is correctly identified as a skos:exactMatch or a skos:closeMatch. We assign 0.5 points if the proposed relation is skos:exactMatch while we consider the relation to be skos:closeMatch, or vice versa. Correspondences between concepts that are not related get 0 points. The scores are used to generate generalized precision and recall figures.

Precision For each participant, we take samples of between 71 and 97 correspondences per GTAA facet for both the GTAA-DBpedia and the GTAA-WordNet alignments and evaluate their correctness in terms of exact match, close match, or no match.

Recall Due to time constraints, we only determined recall of the GTAA Subject facet. We use a small reference alignment from a random sample of 100 GTAA concepts, which we manually mapped to WordNet and DBpedia for the VLCR evaluation of 2008. The result of the GTAA-WordNet and GTAA-DBpedia alignments are compared to the reference alignments.

Inter-rater agreement A team of 4 raters rated random samples of DSSim's correspondences. A team of 3 raters rated the GG2WW correspondences, where each alignment was divided over two raters. One rater was a member of both teams.

In order to check the inter-rater agreement, 100 correspondences were rated by two raters. The agreement was high with a Cohen's kappa of 0.87. In addition, we compared this year's evaluation samples with those of 2008. 120 correspondences appeared in both sets, and again the agreement between the scores was high; Cohen's kappa was 0.92.

10.3 Results

Two teams participated to the OAEI VLCR task: DSSim and GG2WW. Table 27 shows the number of concepts in each resource and the number of correspondences returned for each resource pair. Both participants produced only exact matches. After consulting the participants, we have considered using the confidence measures as an indication of

the strenght of the mapping: a mapping with a confidence measure of 1 was seen as an exact match and a mapping with a confidence measure < 1 was seen as a close match. However, this idea lead to lower precision values for both participants and was therefore abandoned. All correspondences in Table 27 are considered to be exact matches.

GTAA facet	#concepts	#corresp. DSSim		#corresp. GG2WW	
		to WN	to DBp	to WN	to DBp
Subject	3800	655	1363	3663	3381
People	97.000	82	2238	0	17516
Names	27.000	681	3989	0	9023
Locations	14.000	987	5566	0	9527
Total	141.800	2405	13156	3663	39447

Table 27. Number of correspondences in each alignment.

Table 28 shows the precision and recall of both systems.

	Precision		Recall	
Alignment	GG2WW	DSSim	GG2WW	DSSIM
name-dbp	0.63	0.64		
location-dbp	0.94	0.80		
person-dbp	0.91	0.79		
subject-dbp	0.86	0.70	0.62	0.30
name-wn	.	0.44		
location-wn	.	0.61		
person-wn	.	0.07		
subject-wn	0.59	0.77	0.59	0.19
total	0.78	0.62		

Table 28. Precision and recall of DSSim and GG2WW for each GTAA facet-resource pair.

Regarding precision, GG2WW scores consistently better than DSSim on the GTAA-DBpedia alignments. Both systems show a similar pattern when comparing the scores of the four GTAA facets: the scores of the Location facet are highest, followed by the Person, Subject and finally the Name facet. DSSim scores best on the GTAA-WordNet alignments, although a comparison is limited since GG2WW only returned correspondences to the GTAA Subject facet.

DSSim has participated in the VLCR task of 2008 as well. However, a direct comparison of the precision scores of 2008 and 2009 is difficult due to differences in the task; in 2008 we considered SKOS exact-, broad-, narrow- and related-matches. The results of 2008 and 2009 do show similarities when comparing the scores of the facets and resources. The GTAA Names facet remains hard to match, which might be due to the many Dutch-specific concepts in this facet, such as Dutch ships named after famous people. WordNet appears again to be less compatible with the GTAA facets, with the exception of the Subject facet.

Recall measures can be compared to last year directly, as we have used the same evaluation measures and reference alignment. DSSim scores exactly the same on the

GTAA-WordNet mapping (0.19) and higher on the GTAA-DBpedia mapping (from 0.22 to 0.30). GG2WW produced 50% more correspondences between GTAA-WordNet and 300% more correspondences between GTAA-DBpedia than DSSIM (Table 27). This translates to a recall score that is 3 and 2 times as high as the DSSim scores.

11 Structural Preservation Measures

This year we performed analyses of the extent to which particular alignments preserved the structure between two ontologies, or more specifically, between two class hierarchies [15; 5]. Here we provide a brief summary of the approach and presentation of the results.

We wish to measure the *smoothness* of such an alignment, while recognizing that being a smooth mapping is neither necessary nor sufficient to be a good mapping. Nonetheless a strong correlation of smoothness with precision, recall or F-measure promises a potentially *automatic* predictor of alignment quality independent of a reference alignment. Additionally, knowledge of the structural properties of alignments is useful for ontology matchers, especially when providing alignments within one domain where structural preservation is desired.

An alignment is modeled as a relation between two semantic hierarchies, modeled as partially ordered sets [6]. Such ordered structures are not, in general, trees, nor even lattices, but can be rich in multiple inheritance and lack unique least common subsumers between nodes.

Let a semantic hierarchy be a bounded partially ordered set (poset) $\mathcal{P} = \langle P, \leq \rangle$, where P is a finite set of ontology nodes, and $\leq \subseteq P^2$ is a reflexive, anti-symmetric, and transitive binary relation such as subsumption ("`is-a`"). For two taxonomies $\mathcal{P} = \langle P, \leq \rangle, \mathcal{P}' = \langle P', \leq' \rangle$, an alignment relation $F \subseteq P \times P'$ is a collection of pairs $\boldsymbol{f} = \langle a, a' \rangle \in F$, indicating that the node $a \in P$ on the "left" side is mapped or aligned to the node $a' \in P'$ on the "right" side. F determines a domain and codomain

$$Q = \{a \in P, \exists a' \in P', \langle a, a' \rangle \in F\} \subseteq P, \quad Q' = \{a' \in P', \exists a \in P, \langle a, a' \rangle \in F\} \subseteq P',$$

We call the $\boldsymbol{f} \in F$ links, the $a \in Q$ the left anchors and the $a' \in Q'$ the right anchors. Let $m = |Q|, m' = |Q'|$, and $N = |F| \leq mm'$.

Our approach is not a *relative* measure of an alignment with respect to a reference alignment, but rather an *inherent* or *independent* measure of the alignment based on the following principles:

Twist, or order discrepancy: a, b should have the same structural relations in $\mathcal{P}$ as a', b' in $\mathcal{P}'$

Stretch, or distance discrepancy: Relative distance between $a, b \in P$ should be the same as $a', b' \in P'$

Let d be a metric on $\mathcal{P}$ and $\mathcal{P}'$. For links $\boldsymbol{f} = \langle a, a' \rangle, \boldsymbol{g} = \langle b, b' \rangle \in F$, we want the metric relations between the $a, b \in Q$ to be the same as their corresponding $a', b' \in Q'$, so that $|\bar{d}(a, b) - \bar{d}'(a', b')|$ is small. In this work, we use the upper and lower cardinality-based distances:

$$d_u(a, b) = |\uparrow a| + |\uparrow b| - 2 \max_{c \in a \vee b} |\uparrow c|, \qquad d_l(a, b) = |\downarrow a| + |\downarrow b| - 2 \max_{c \in a \wedge b} |\downarrow c|,$$

where for a node $a \in P$, its upset $\uparrow a = \{x | x \geq a\}$ and downset $\downarrow a = \{x | x \leq a\}$ are all its ancestors and successors respectively, so that $|\uparrow a|, |\downarrow a|$ are the number of ancestors and successors. The generalized join and meet are

$$a \vee b = \text{Min}(\uparrow a \bigcap \uparrow b) \subseteq P, \qquad a \wedge b = \text{Max}(\downarrow a \bigcap \downarrow b) \subseteq P,$$

where for a set of nodes $R \subseteq P$ the upper bounds and lower bounds are

$$\text{Min}(R) = \{a \in R : \nexists b \in R, b < a\} \subseteq P, \text{Max}(R) = \{a \in R : \nexists b \in R, b > a\} \subseteq P.$$

We need to measure the relative proportion of the overall structure two nodes are apart, so define the normalized upper and lower distances as:

$$\bar{d}_u(a, b) = \frac{d_u(a, b)}{|P| - 1} \in [0, 1], \quad \bar{d}_l(a, b) = \frac{d_l(a, b)}{|P| - 1} \in [0, 1].$$

Let d be a metric used in both $\mathcal{P}, \mathcal{P}'$, in our case, the lower distance d_l. Then the link discrepancy is given by: $\delta(\boldsymbol{f}, \boldsymbol{g}) = |\bar{d}(a, b) - \bar{d}(a', b')|$, and the distance discrepancy induced by F between $\mathcal{P}$ and $\mathcal{P}'$ given d is:

$$D(F) = \frac{\sum_{\boldsymbol{f}, \boldsymbol{g} \in F} \delta(\boldsymbol{f}, \boldsymbol{g})}{\binom{N}{2}}.$$

$D(F) \in [0, 1]$, with $D(F) = 0$ iff F is completely distance preserving, and $D = 1$ if F is maximally distance distorting, e.g. mapping diameters to equality, and neighbors and children to diameters. We also calculate the order discrepancy of each alignment as:

$$\Gamma(F) = \frac{\sum_{\boldsymbol{f}, \boldsymbol{g} \in F} \gamma(\boldsymbol{f}, \boldsymbol{g})}{\binom{N}{2}},$$

where for a pair of links $\boldsymbol{f}, \boldsymbol{g} \in F$, and a relation $* \in \{<, >, =, \not\sim\}$ ($\not\sim$ denoting non comparability),

$$\gamma(\boldsymbol{f}, \boldsymbol{g}) = \begin{cases} 0, & \text{if } a * b \text{ and } a' * b' \\ 1, & \text{otherwise} \end{cases}$$

Hence $D(F)$ measures the "stretching" of F, $\Gamma(F)$ measures "twisting", or the number of purely structural violations present.

Figure 13, Figure 14, and Figure 15 show scatter plots of $D(F)$ against precision for all the 1xx, 2xx, and 3xx tests for the benchmark track, respectively. We see a moderate trend of decreasing precision with increasing $D(F)$, with Pearson correlation coefficients of $r = -0.65$ and $r = -0.51$ respectively. Table 29 shows the correlation r for $D(F)$ and $\Gamma(F)$ against precision, recall, and F-measure for all tracks, and all 1xx, 2xx, and 3xx tracks grouped together.

For more details on a particular track, Table 30 shows the results from Test 205 from Benchmark. We can see in this case a particular strong dropoff in precision with increasing discrepancy, with $r = -0.92$.

Table 31 shows the results for the anatomy track. Scatter plots are shown in Figure 16 for all tests. Table 32 summarizes the correlations, combining all tests, and then

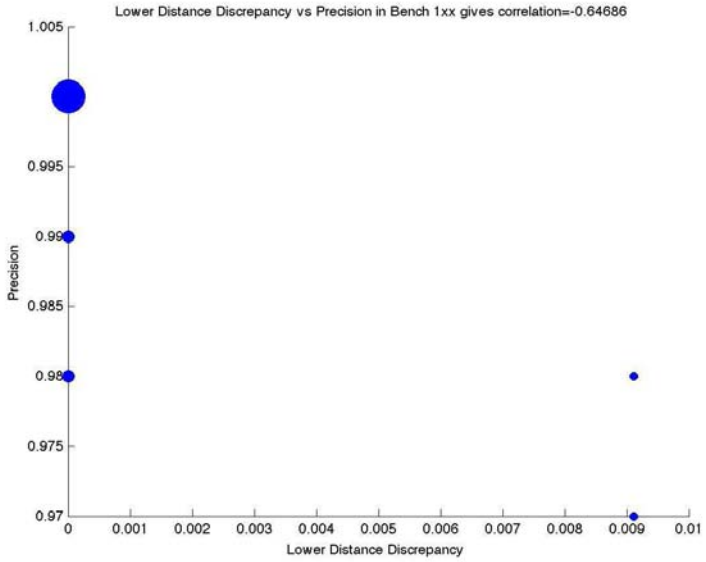


Fig. 13. Precision vs. $D(F)$, Benchmark track: 1xx tests. Marker size is proportional to the number of tests with that combination of values.

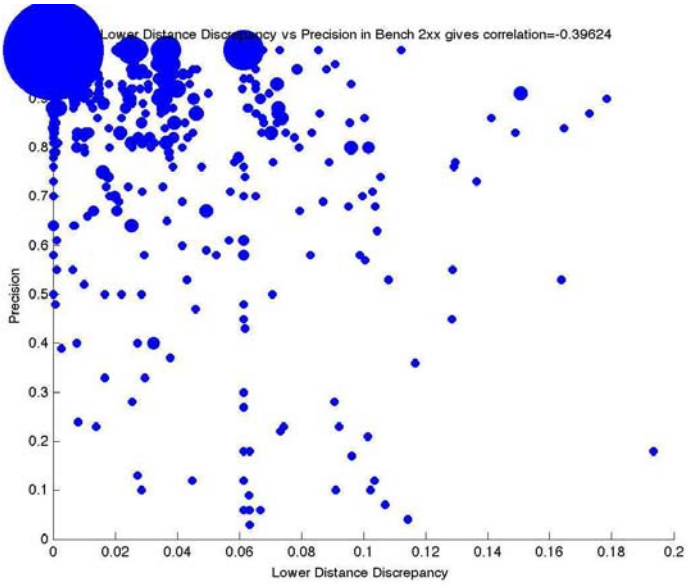


Fig. 14. Precision vs. $D(F)$, Benchmark track: 2xx tests.

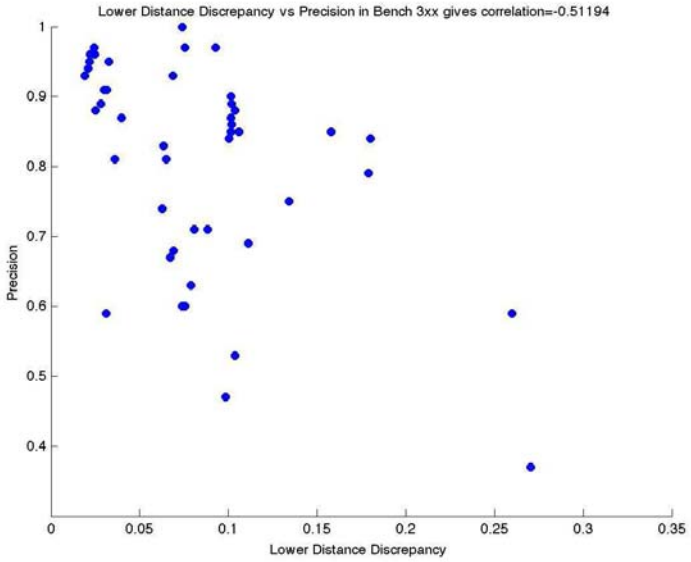


Fig. 15. Precision vs. $D(F)$, Benchmark track: 3xx tests.

r		Prec.	Rec.	FMeas.
Bench 1*	$D(F)$	-0.65	0.03	0.02
	$\Gamma(F)$	-0.65	0.03	0.02
Bench 2*	$D(F)$	-0.41	-0.13	-0.18
	$\Gamma(F)$	-0.48	-0.25	-0.29
Bench 3*	$D(F)$	-0.51	-0.48	-0.53
	$\Gamma(F)$	-0.54	-0.39	-0.46
Bench [1-3]*	$D(F)$	-0.39	-0.02	-0.07
	$\Gamma(F)$			-0.20

Table 29. Pearson correlations for $D(F)$ and $\Gamma(F)$ against precision, recall and F-measure for all tracks, and all 1xx, 2xx, and 3xx tracks grouped together.

Submitter	$D(F)$	Prec.	Rec.	FMeas.
$r(D(F), \cdot)$:		-0.92	-0.73	-0.76
refalign	0.0	1.0	1.0	1.0
MapPSO	0.0	1.0	0.99	0.99
AROMA	0.0	1.0	0.99	0.99
ASMOV	0.0	1.0	0.99	0.99
Lily	0.0	1.0	0.99	0.99
RiMOM	0.0	1.0	0.99	0.99
GeRoMe	0.0	1.0	0.97	0.98
AgrMaker	0.0	1.0	0.97	0.98
DSSim	0.0	0.91	0.81	0.86
aflood	0.008	0.91	0.75	0.82
kosimap	0.083	0.83	0.59	0.69
SOBOM	0.0	1.0	0.29	0.45
TaxoMap	0.108	0.53	0.09	0.15

Table 30. Benchmark 205 results.

broken out by test. Again, we see a strong correlation of increasing $D(F)$ against especially decreasing precision. Note the outlier point, corresponding to Taxomap in test 3 with $D(F) = 0.00145$. If this point is excluded, then among all tests we obtain r values of -0.84 for precision, 0.05 for recall, and -0.61 for F-measure.

These preliminary results are clearly in need of further analysis, which we are now embarking on. Some early comments include:

- These results are consistent with those shown in [5], which showed a moderate correlation of $D(F)$ with F-measure.
- Pearson correlation, the only measure here, is a weak indicator, but suggestive that our lower distance discrepancy may act as a predictor of precision.
- Here only the lower distance $d_l(a, b)$ and distance discrepancy $D(F)$ were used. Further consideration is also required of the role the upper distance $d_u(a, b)$ and the order discrepancy $\Gamma(F)$.

Test	Submitter	$D(F)$	$\Gamma(F)$	Prec.	Rec.	FMeas.
1	aflood	0.00133	0.00155	0.873	0.653	0.747
1	AgrMaker	0.00127	0.00147	0.865	0.798	0.831
1	AROMA	0.00288	0.00298	0.775	0.678	0.723
1	ASMOV	0.00314	0.00368	0.746	0.755	0.751
1	DSSim	0.00156	0.00233	0.853	0.676	0.754
1	kosimap	0.00099	0.00123	0.866	0.619	0.722
1	Lily	0.00259	0.00346	0.738	0.739	0.739
1	Ref_Full	0.00078	0.00066	1.0	1.0	1.0
1	SOBOM	0.00088	0.00091	0.952	0.777	0.855
1	taxomap	0.00149	0.00225	0.87	0.678	0.762
2	aflood	0.00105	0.00098	0.892	0.712	0.792
2	AgrMaker	0.00086	0.00081	0.967	0.682	0.8
2	ASMOV	0.00133	0.00161	0.821	0.736	0.776
2	DSSim	0.00113	0.00123	0.973	0.62	0.757
2	kosimap	0.0023	0.00443	0.907	0.446	0.598
2	Lily	0.00236	0.00341	0.869	0.559	0.681
2	taxomap	0.00075	0.00086	0.953	0.609	0.743
3	aflood	0.00148	0.0016	0.827	0.763	0.794
3	AgrMaker	0.00332	0.00368	0.511	0.815	0.628
3	ASMOV	0.00306	0.00386	0.725	0.767	0.745
3	kosimap	0.00099	0.00123	0.866	0.619	0.722
3	Lily	0.00332	0.00393	0.534	0.774	0.632
3	taxomap	0.01486	0.02115	0.458	0.716	0.559
4	aflood	0.00145	0.00155			
4	AgrMaker	0.00077	0.00066			
4	ASMOV	0.00373	0.0041			
4	taxomap	0.00474	0.00748			

Table 31. Results of anatomy track.

r		Prec.	Rec.	FMeas.
Anatomy 1	$D(F)$	-0.91	-0.25	-0.55
	OD	-0.94	-0.35	-0.64
Anatomy 2	$D(F)$	-0.47	-0.71	-0.85
	OD	-0.36	-0.82	-0.94
Anatomy 3	$D(F)$	-0.68	-0.03	-0.76
	OD	-0.65	-0.07	-0.74
Anatomy [1-3]	$D(F)$	-0.73	0.04	-0.59
	OD	-0.69	-0.03	-0.61

Table 32. Pearson correlation r for $D(F)$ and $\Gamma(F)$ against precision, recall and F-measure for all Anatomy tests.

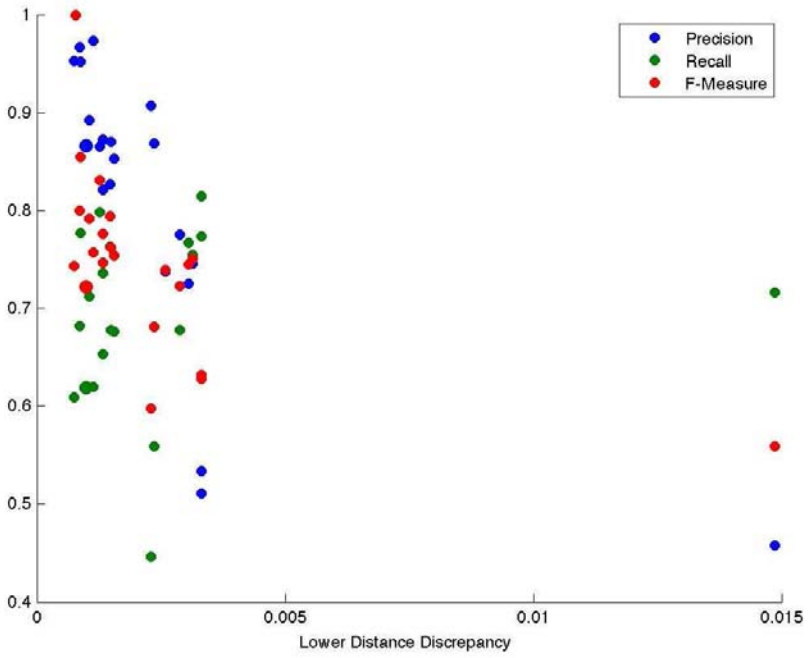


Fig. 16. Anatomy track, all tests, lower distance discrepancy vs. (blue) precision (green) recall (red) F-measure.

12 Lesson learned and suggestions

The lessons learned for this year are relatively similar to those of previous years. There remain one lesson not really taken into account that we identify with an asterisk (*). We reiterate those lessons that still apply with new ones:

A) Unfortunately, we have not been able to maintain the better schedule of two years ago. We hope to be able to improve this through the use of SEALS technology (see §13).
B) The trend that there are more matching systems able to enter such an evaluation seems to slow down. There have been not many new systems this year but on specialised topics. There can be two explanations: the field is shrinking or the entry ticket is too high.
C) We still can confirm that systems that enter the campaign for several times tend to improve over years.
*D) The benchmark test case is not discriminant enough between systems and, as noted last year, automatic test generation could contribute to improve the situation. We plan to introduce this in the SEALS platform.
E) Some tracks provide non conclusive results, we should make effort to improve this situation by knowing, beforehand, what conclusions can be drawn from the evaluations.
F) With the increase in the number of data sets, comes less participants. We will have to set rules for declaring unfruitful, tracks in which there is no minimal independent participation.

Of course, these are only suggestions that will be refined during the coming year, see [22] for a detailed discussion on the ontology matching challenges.

13 Future plans

In order to improve the organization of the Ontology Alignment Evaluation Initiative, plans are made for next year that the evaluation campaign be run on a new open platform for semantic technology evaluation developed by the SEALS project[21]. The SEALS project aims at providing support for the evaluation of semantic technologies, including ontology matching.

The project will provide an automated test infrastructure and will organize integrated evaluation campaigns. This will allow new features in tests cases like test generation on demand and online evaluation. This will lead to a more automated and integrated way to evaluate systems as well as the opportunity for participants to run the evaluation for themselves.

We plan to run the next OAEI campaign within this framework and to have at least three tracks, and if possible more, fully supported by the SEALS platform.

[21] http://www.seals-project.eu

14 Conclusions

Confirming the trend of last year, the number of systems, and tracks they enter in, seems to stabilize. As noticed the previous years, systems which do not enter for the first time are those which perform better. This shows that, as expected, the field of ontology matching is getting stronger (and we hope that evaluation has been contributing to this progress).

Moreover, we had this year more tracks but participants did not enter more tracks than previous years: 3.25 against 3.84 in 2008 and 2.94 in 2007. This figure of around 3 out of 8 may be the result of either the specialization of systems or the short time allowed to the campaign.

All participants have provided a description of their systems and their experience in the evaluation. These OAEI papers, like the present one, have not been peer reviewed. However, they are full contributions to this evaluation exercise and reflect the hard work and clever insight people put in the development of participating systems. Reading the papers of the participants should help people involved in ontology matching to find what makes these algorithms work and what could be improved. Sometimes participants offer alternate evaluation results.

The Ontology Alignment Evaluation Initiative will continue these tests by improving both test cases and testing methodology for being more accurate. Further information can be found at:

`http://oaei.ontologymatching.org`.

Acknowledgments

We warmly thank each participant of this campaign. We know that they have worked hard for having their results ready and they provided insightful papers presenting their experience. The best way to learn about the results remains to read the following papers.

We thank Paolo Bouquet and the OKKAM European Project for providing the reference alignment for the IIMB benchmark used in the instance matching track.

We thank Patrice Landry, Genevieve Clavel and Jeroen Hoppenbrouwers for the MACS data. For LCSH, RAMEAU and SWD, respectively, The Library of Congress, The French National Library and the German National Library. The collection of the British Library was provided by the The European Library Office.

Jérôme Euzenat, Christian Meilicke, Heiner Stuckenschmidt and Cassia Trojahn dos Santos have been partially supported by the SEALS (IST-2009-238975) European project.

We are grateful to Dominique Ritze (University of Mannheim) for participating in extension of reference alignment for the conference track. In addition, Ondřej Šváb-Zamazal and Vojtěch Svátek were supported by the IGA VSE grant no.20/08 “Evaluation and matching ontologies via patterns”.

We also warmly thanks Claudio Baldassarre for preparing unfruitful test cases which were cancelled; we hope to have more success with these in the coming years.

We are grateful to Martin Ringwald and Terry Hayamizu for providing the reference alignment for the anatomy ontologies.

We gratefully acknowledge the Dutch Institute for Sound and Vision for allowing us to use the GTAA. We would like to thank Willem van Hage for the use of his tools for manual evaluation of correspondences.

We also thank the other members of the Ontology Alignment Evaluation Initiative Steering committee: Wayne Bethea (John Hopkins University, USA), Lewis Hart (AT&T, USA), Tadashi Hoshiai (Fujitsu, Japan), Todd Hughes (DARPA, USA), Yannis Kalfoglou (Ricoh laboratories, UK), John Li (Teknowledge, USA), Miklos Nagy (The Open University (UK), Natasha Noy (Stanford University, USA), Yuzhong Qu (Southeast University (China), York Sure (Leibniz Gemeinschaft, Germany), Jie Tang (Tsinghua University (China), Raphaël Troncy (Eurecom, France), and Petko Valtchev (Université du Québec Montréal, Canada).

References

1. Zharko Aleksovski, Warner ten Kate, and Frank van Harmelen. Exploiting the structure of background knowledge used in ontology matching. In *Proc. 1st International Workshop on Ontology Matching (OM-2006), collocated with ISWC-2006*, Athens, Georgia (USA), 2006.
2. Ben Ashpole, Marc Ehrig, Jérôme Euzenat, and Heiner Stuckenschmidt, editors. *Proceedings of the K-Cap Workshop on Integrating Ontologies*, Banff (CA), 2005.
3. Oliver Bodenreider, Terry Hayamizu, Martin Ringwald, Sherri De Coronado, and Songmao Zhang. Of mice and men: Aligning mouse and human anatomies. In *Proc. American Medical Informatics Association (AIMA) Annual Symposium*, pages 61–65, 2005.
4. Caterina Caracciolo, Jérôme Euzenat, Laura Hollink, Ryutaro Ichise, Antoine Isaac, Véronique Malaisé, Christian Meilicke, Juan Pane, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, and Vojtech Svátek. Results of the ontology alignment evaluation initiative 2008. In *Proc. 3rd International Workshop on Ontology Matching (OM-2008), collocated with ISWC-2008*, Karlsruhe (Germany), 2008.
5. Joslyn Cliff, Paulson Patrick, and White Amanda. Measuring the structural preservation of semantic hierarchy alignments. In *Proc. 4th International Workshop on Ontology Matching (OM-2009), collocated with ISWC-2009*, Chantilly (USA), 2009. this volume.
6. Brian Davey and Hilary Priestly. *Introduction to lattices and order*. Cambridge University Press, Cambridge, 2nd edition, 1990.
7. Marc Ehrig and Jérôme Euzenat. Relaxed precision and recall for ontology matching. In *Proceedings of the K-Cap Workshop on Integrating Ontologies*, pages 25–32, Banff (CA), 2005.
8. Jérôme Euzenat. An API for ontology alignment. In *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, pages 698–712, Hiroshima (JP), 2004.
9. Jérôme Euzenat, Malgorzata Mochol, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab, Vojtech Svatek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2006. In *Proc. 1st International Workshop on Ontology Matching (OM-2006), collocated with ISWC-2006*, pages 73–95, Athens, Georgia (USA), 2006.
10. Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, Heidelberg (DE), 2007.
11. Jérôme Euzenat, Antoine Isaac, Christian Meilicke, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svab, Vojtech Svatek, Willem Robert van Hage, and Mikalai Yatskevich. Results of the ontology alignment evaluation initiative 2007. In *Proc. 2nd International Workshop on Ontology Matching (OM-2008), collocated with ISWC-2007*, pages 96–132, Busan (Korea), 2007.

12. Daniel Fleischhacker and Heiner Stuckenschmidt. Implementing semantic precision and recall. In *Proc. 4th International Workshop on Ontology Matching (OM-2009), collocated with ISWC-2009*, Chantilly (USA), 2009. this volume.
13. Fausto Giunchiglia, Mikalai Yatskevich, Paolo Avesani, and Pavel Shvaiko. A large scale dataset for the evaluation of ontology matching systems. *The Knowledge Engineering Review Journal*, 24(2):137–157, 2009.
14. Antoine Isaac, Henk Matthezing, Lourens van der Meij, Stefan Schlobach, Shenghui Wang, and Claus Zinn. Putting ontology alignment in context: Usage scenarios, deployment and evaluation in a library case. In *Proceedings of the 5th European Semantic Web Conference (ESWC)*, pages 402–417, Tenerife (ES), 2008.
15. Cliff Joslyn, Alex Donaldson, and Patrick Paulson. Evaluating the structural quality of semantic hierarchy alignments. In *International Semantic Web Conference (Posters & Demos)*, Karlsruhe (Germany), 2008.
16. Patrick Lambrix and Qiang Liu. Using partial reference alignments to align ontologies. In *Proceedings of the 6th European Semantic Web Conference*, pages 188–202, Heraklion, Crete (Greece), 2009.
17. Christian Meilicke and Heiner Stuckenschmidt. Incoherence as a basis for measuring the quality of ontology mappings. In *Proc. 3rd International Workshop on Ontology Matching (OM-2008), collocated with ISWC-2008*, pages 1–12, Karlsruhe (Germany), 2008.
18. Christian Meilicke and Heiner Stuckenschmidt. An efficient method for computing a local optimal alignment diagnosis. Technical report, University Mannheim, Computer Science Institute, 2009.
19. Vojtěch Svátek Ondřej Šváb-Zamazal O. Empirical knowledge discovery over ontology matching results. In *Proc. 1st ESWC International Workshop on Inductive Reasoning and Machine Learning on the Semantic Web*, Heraklion (Greece), 2009.
20. Marta Sabou, Mathieu d'Aquin, and Enrico Motta. Using the semantic web as background knowledge for ontology mapping. In *Proc. 1st International Workshop on Ontology Matching (OM-2006), collocated ISWC-2006*, pages 1–12, Athens, Georgia (USA), 2006.
21. Francois Scharffe. *Correspondence Patterns Representation*. PhD thesis, University of Innsbruck, 2009.
22. Pavel Shvaiko and Jérôme Euzenat. Ten challenges for ontology matching. In *Proceedings of the 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, pages 1164–1182, Monterrey (MX), 2008.
23. York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proceedings of the ISWC Workshop on Evaluation of Ontology-based Tools (EON)*, Hiroshima (JP), 2004.
24. Willem Robert van Hage, Antoine Isaac, and Zharko Aleksovski. Sample evaluation of ontology-matching systems. In *Proc. 5th International Workshop on Evaluation of Ontologies and Ontology-based Tools (EON 2007), collocated with ISWC-2007*, pages 41–50, Busan (Korea), 2007.

Grenoble, Milano, Amsterdam, Richland, Mannheim, Milton-Keynes, Samos, Trento, Prague, November 2009

Anchor-Flood: Results for OAEI 2009

Md. Hanif Seddiqui and Masaki Aono

Toyohashi University of Technology, Japan
hanif@kde.ics.tut.ac.jp, aono@ics.tut.ac.jp

Abstract. Our ontology schema matching algorithm takes the essence of the *locality of reference* by considering the neighboring concepts and relations to align the entities of ontologies. It starts off a seed point called an *anchor* (a pair of "look-alike" concepts across ontologies) and collects two blocks of neighboring concepts across ontologies. The concepts of the pair of blocks are aligned and the process is repeated for newly found aligned pairs. This year, we use a semantically reformed dynamic block of concepts starting from an anchor-concept and produce two blocks from one anchor to get alignment. We improve our memory management. The experimental results show its effectiveness against the benchmark, anatomy track and other datasets. We also extend our algorithm to match instances of IIMB benchmarks and we obtained effective results.

1 Presentation of the system

During OAEI-2008, our ontology alignment system used the *locality of reference* for collecting neighboring concepts with strong semantic arbitrary depth for aligning concepts across pair of ontologies. This year, we incorporate a process of collecting concepts with strong intrinsic semantic similarity within ontology elements considering intrinsic Information Content (IC) [6] to form a dynamic block. Hence our system forms a pair of dynamic blocks staring off an anchor across ontologies. We improve our memory management to cope large scale ontology alignment effectively. Our algorithm has shorter run time than that of the previous year. It takes less memory and even less time as well to align large ontologies. We participate in the benchmark datasets, all four tasks of anatomy track, conference and directory as well. We also take limited participation in the instance matching track. We participate only in the IIMB benchmark track of instance matching track.

1.1 State, purpose, general statement

The purpose of our Anchor-Flood algorithm [8] is basically ontology matching. However, we use our algorithm in patent mining system to classify a research abstract in terms of International Patent Classification (IPC). Containing mostly general terminologies in an abstract leads classification to a formidable task. Automatic extracted taxonomy of related terms available in an abstract is aligned with the

taxonomy of IPC ontology with our algorithm successfully.
Furthermore, we use our algorithm to integrate the multimedia resources represented by MPEG-7 [5] ontologies [11]. We have achieved good performance with effective results in the field of multimedia resource integration [7].
To be specific, we describe our Anchor-Flood algorithm, instance matching algorithm and their results against OAEI 2009 datasets here.

1.2 Specific techniques used

We have two parts of our system. One is the ontology schema matching Anchor-Flood algorithm to align concepts and properties of a pair of ontologies. Another is the instance matching approach which essentially uses our Anchor-Flood algorithm. We implement our system in Java. We create our own memory model of ontology by the ARP triple parser of Jena module.

1.2.1 Ontology Schema Matching

As a part of preprocessing, our system parses ontologies into our own developed memory model by using ARP triple parser of Jena. We also normalize the lexical description of ontology entities.

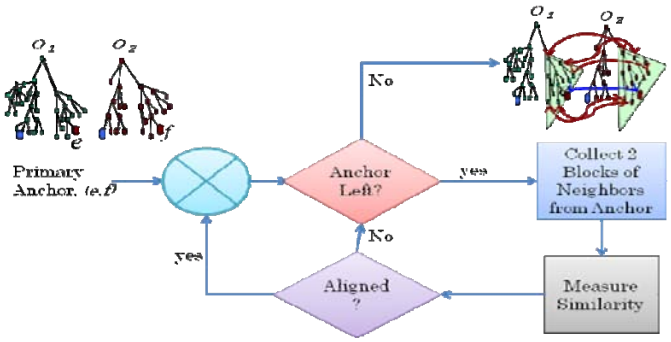


Fig.1. Ontology schema matching Anchor-Flood algorithm

Our schema matching algorithm starts off an anchor. It has a complex process of collecting small blocks of concepts and related properties dynamically by considering super-concept, sub-concept, siblings and few other neighbors from the anchor point. The size of blocks affect the running time adversely. Therefore, we incorporate semantic similarity considering intrinsic Information Content (IC) for building blocks of neighboring concepts from anchor-concepts.
Local alignment process aligns concepts and their related properties based on lexical information [2, 10, and 12], and structural relations [1, 3, 4]. Retrieved aligned pairs are considered as anchors for further processing. The process is repeated until there is no more aligned pair to be processed. Hence, it burst out with a pair of aligned fragment of the ontologies, giving the taste of segmentation [9]. Multiple anchors from different part of ontologies confirm a fair collection of aligned pairs as a whole.

1.2.2 Ontology Instance Matching

In an ontology, neither a concept nor an instance comprises its full specification in its name or URI alone. Therefore we consider the semantically linked information that includes linked concepts, properties and their values and other instances as well. They all together make an information cloud to specify the meaning of that particular instance. We refer this collective information of association as *Semantic Link cloud.* The degree of certainty is proportional to the number of semantic link associated to a particular instance by means of property values and other instances. First, pair of TBox is aligned with our Anchor-Flood algorithm. Then, we check the alignment of the type of an instance to any concept of the neighbors of the type of another instance across ABox. We measure the structural similarity among the elements available in a pair of clouds to produce instance alignment. The instance matching algorithm is depicted in Fig. 2 and in Fig. 3.

Alg. *InstanceMatch* (ABox ab_1, ABox ab_2, Alignment A)

1. for each $ins_i \in ab_1$
2. $\quad cloud_i = makeCloud(ins_i, ab_1)$
3. $\quad$ for each $ins_j \in ab_2$
4. $\quad\quad cloud_j = makeCloud(ins_j, ab_2)$
5. $\quad\quad$ if $\exists a(c_1, c_2) \in A | c_1 \in Block(ins_1.type) \wedge c_2 \in Block(ins_2.type)$
6. $\quad\quad\quad$ if $Sim_{struct}(cloud_i, cloud_j) \geq \delta$
7. $\quad\quad\quad\quad imatch = imatch \cup makeAlign(ins_i, ins_j)$

Fig. 2 Pseudo code of instance matching algorithm

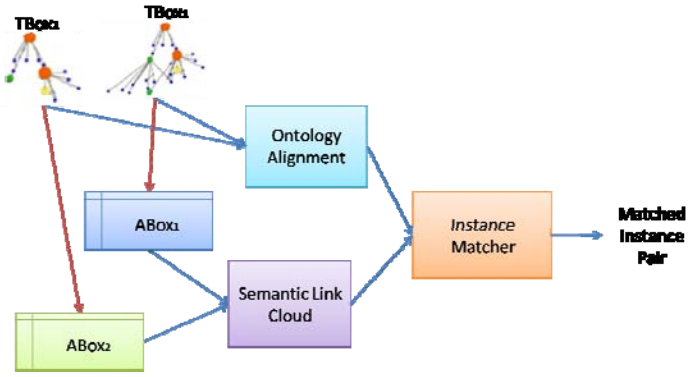


Fig. 3 The basic block diagram of our instance matching approach

1.3 Adaptations made for the evaluation

The Anchor-Flood algorithm needs an anchor to start off. Therefore, we use a tiny program module for extracting probable aligned pairs as anchors. It uses lexical information and some statistical information to extract a small number of aligned pairs from different part of ontologies. The program is essentially smaller, simpler

and faster. We also removed the subsumption module of our algorithm to keep it faster.

1.4 Link to the system and parameters file

The version of Anchor-Flood for OAEI-2009 can be downloaded from our website: http://www.kde.ics.tut.ac.jp/~hanif/res/2009/anchor_flood.zip. The parameter file is also included in the anchor_flood.zip file. I recommend readers to read the readme.txt file first. The file includes the necessary description and parameters as well in brief.

1.5 Link to the set of provided alignments (in align format)

The results for OAEI-2008 are available at our website:
http://www.kde.ics.tut.ac.jp/~hanif/res/2009/aflood.zip.

2 Results

In this section, we describe the results of Anchor-Flood algorithm against the benchmark, anatomy, directory and conferences ontologies and the IIMB instance matching benchmark provided by the OAEI 2009 campaign.

2.1 benchmark

On the basis of the nature, we can divide the benchmark dataset into five groups: #101-104, #201-210, #221-247, #248-266 and #301-304. We describe the performance of our Anchor-Flood algorithm over each of the groups below:
#101-104. Table 1 shows the perfect precision and recall in this group.
#201-210. We improve our results in this group compared to last year results as we improve our structural similarity measure.
#221-247. Our algorithm produces good precision and recall as the previous year.
#248-266. This is the most difficult group for our algorithm. However, we improve our result compared to the last year.
#301-304 Our algorithm produce almost similar result as the previous year.

Table 1. Average results against the ontology benchmarks

Datasets	Prec.	Rec.	F-Measure
101-104	1.00	1.00	1.00
201-210	0.99	0.97	0.98
221-247	0.99	1.00	0.99
248-266	0.96	0.73	0.83
301-304	0.88	0.77	0.82

2.2 anatomy

In this test, the real world cases of anatomy for Adult Mouse Anatomy (2744 classes) and NCI Thesaurus (3304 classes) for human anatomy are included. These are relatively large compared to benchmark ontologies. We participated all of the tasks of this track this year. Our algorithm produces similar result four times faster than the last year. We participate in task#2, task#3 and task#4 for the first time. We find that the run time changes adversely if the block size increases.

Table 2. Our algorithm collects alignment from anatomy ontologies quickly.

Task	**Description**	**Required Time (sec)**	**|Total Alignment|**
Task#1	Default Optimization	14.5	1149
Task#2	Increase precision	221	1228
Task#3	Increase recall	278	1416
Rask#4	Extended reference mapping	282	1460

2.3 directory & Conference Tracks

We also participate directory and conference track this year for the first time.

2.4 Instance Matching: IIMB Benchmarks

On the basis of transformation, the benchmark dataset is divided into four groups: 001-010, 011-019, 020-029 and 030-037. Table 3 shows the precision and recall for each of the groups. However, the detailed results are displayed in Annex section of this paper.

Table 3. Instance matching results against IIMB benchmarks

Datasets	**Trnasformation**	**Prec.**	**Rec.**	**F-Measure**
001-010	Value transformations	0.99	0.99	0.991
011-019	Structural transformations	0.72	0.79	0.751
020-029	Logical transformations	1.00	0.96	0.981
030-037	Several combinations of the previous transformations	0.75	0.82	0.786

3 General comments

In this section, we want to comment on the results of our system and the way to improve it.

3.1 Comments on the results

The main strength of our schema matching system is the way of minimizing the comparisons between entities, which leads enhancement in running time. In instance matching, our system shows its strength over value and logical transformations.
The weak points are: our system ignores some distantly placed aligned pairs in ontology alignment system. In instance matching, we have still rooms to work in structural transformation.

3.2 Discussions on the way to improve the proposed system

It has still rooms of improving alignments strengthening the semantic and structural analysis and adding background knowledge. We also want to incorporate complex alignment like subsumption and 1:n alignments. In instance matching, we want to improve our system against structural transformation.

4 Conclusion

Ontology matching is very important for attaining interoperability as the core of every semantic application is ontology. We implemented faster algorithm to align specific interrelated parts across ontologies, which gives the flavor of segmentation. The anatomical ontology matching shows the effectiveness of our Anchor-Flood algorithm. Our instance matching algorithm also shows its strength in value and logical transformations. In structural transformation our algorithm is also effective in spite of challenging transformation. We improved our previous Anchor-Flood algorithm in several perceptions to retrieve ontology alignment. Furthermore, we improve the versatility of using it in different applications including instance matching, patent classification and multimedia resource integration.

References

1. **Bouquet, P., Serafini, L. and Zanobini, S.:** *Semantic Coordination: A New Approach and an Application.* Proceedings of the 2nd International Semantic Web Conference (ISWC2003), Sanibel Island, Florida, USA (2003) pp. 130-145.
2. **Euzenat, J. and Valtchev, P.:** *Similarity-based Ontology Alignment in OWL-Lite.* Proceedings of the 16th European Conference on Artificial Intelligence (ECAI2004), Valencia, Spain (2004) pp. 333-337.
3. **Giunchiglia, F. and Shaiko, P.:** *Semantic Matching,* The Knowledge Engineering Review, Cambridge Univ Press, Vol. 18(3), 2004, pp. 265-280.
4. **Giunchiglia, F., Shvaiko, P. and Yatskevich, M.:** *S-Match: an Algorithm and an Implementation of Semantic Matching.* Proceedings of the 1st European Semantic Web Symposium (ESWS2004), Heraklion, Greece, (2004) pp. 61-75.
5. **Nack, F. and Lindsay, A.T.:** *Everything you wanted to know about MPEG-7 (Part I).* IEEE Multimedia, Vol. 6(3), 1999, pp. 65--77.

6. **Resnik, P.:** *Using information content to evaluate semantic similarity in a taxonomy.* Proceedings of the 14th International Joint Conference on Artificial Intelligence. Montreal, Canada (1995) pp. 448-453.
7. **Seddiqui, M.H. and Aono, M.:** *MPEG-7 based Multimedia Information Integration through Instance Matching.* Berkeley, IEEE International Conference on Semantic Computing, CA, USA (2009) pp. 618-623.
8. **Seddiqui, M.H. and Aono, M.:** An Efficient and Scalable Algorithm for Segmented Alignment of Ontologies of Arbitrary Size. *Web Semantics: Science, Services and Agents on the World Wide Web* (2008), doi:10.1016/j.websem.2009.09.001.
9. **Seidenberg, J. and Rector, A.:** *Web Ontology Segmentation: Analysis, Classification and Use.* Proceedings of the 15th International Conference on World Wide Web (WWW2006), Edinburgh, Scotland (2006) pp. 13-22.
10. **Stoilos, G., Stamou, G. and Kollias, S.:** *A String Metric for Ontology Alignment.* Proceedings of the 4th International Semantic Web Conference (ISWC2005), Galway, Ireland (2005) pp. 623-637.
11. **Troncy, R., et al.:** *Mpeg-7 based Multimedia Ontologies: Interoperability Support or Interoperability Issue.* Proceedings of the 1st International Workshop on Multimedia Annotation and Retrieval enabled by Shared Ontologies (MAReSO), Genova, Italy (2007).
12. **Winkler, W.E.:** *The State of Record Linkage and Current Research Problems.* Technical Report, Statistical Research Division, U.S. Census Bureau, Washington, USA (1999).

Annex

Schema Matching: Ontology Benchmark

Dataset	Prec.	Rec.	F-Meas.	Time (ms)
101	1.00	1.00	1.00	518
103	1.00	1.00	1.00	155
104	1.00	1.00	1.00	157
201	0.95	0.90	0.92	160
201-2	1.00	1.00	1.00	165
201-4	1.00	1.00	1.00	155
201-6	0.98	0.98	0.98	154
201-8	0.98	0.97	0.97	177
202	1.00	0.97	0.98	125
202-2	1.00	1.00	1.00	121
202-4	1.00	1.00	1.00	141
202-6	1.00	1.00	1.00	128
202-8	1.00	0.98	0.99	135
203	1.00	1.00	1.00	131
204	0.99	0.98	0.98	139
205	0.92	0.85	0.88	156
206	1.00	0.97	0.98	171
207	1.00	0.97	0.98	156
208	0.99	0.98	0.98	120
209	0.93	0.82	0.87	143
210	1.00	0.96	0.98	132
221	1.00	1.00	1.00	125
222	1.00	1.00	1.00	151
223	1.00	1.00	1.00	138
224	1.00	1.00	1.00	112
225	1.00	1.00	1.00	134
228	1.00	1.00	1.00	73
230	0.94	1.00	0.97	119
231	1.00	1.00	1.00	127
232	1.00	1.00	1.00	119
233	1.00	1.00	1.00	66
236	1.00	1.00	1.00	62
237	1.00	1.00	1.00	117
238	1.00	1.00	1.00	132
239	0.97	1.00	0.98	74
240	0.94	0.97	0.95	77
241	1.00	1.00	1.00	71
246	0.97	1.00	0.98	64
247	0.94	0.97	0.95	79
248	1.00	0.61	0.76	108
248-2	1.00	0.97	0.98	123
248-4	1.00	0.96	0.98	110
248-6	1.00	0.90	0.95	107
248-8	1.00	0.78	0.88	108
249	1.00	0.78	0.88	103
249-2	1.00	1.00	1.00	105
249-4	1.00	1.00	1.00	106
249-6	1.00	1.00	1.00	122
249-8	1.00	0.98	0.99	65
250	1.00	1.00	1.00	63
250-2	1.00	1.00	1.00	63
250-4	1.00	1.00	1.00	79
250-6	1.00	1.00	1.00	66
250-8	1.00	0.97	0.98	119
251	1.00	0.37	0.54	131
251-2	1.00	0.92	0.96	136
251-4	0.98	0.85	0.91	136

251-6	0.97	0.74	0.84	128
251-8	1.00	0.62	0.77	136
252	0.97	0.29	0.45	129
252-2	0.98	0.92	0.95	132
252-4	0.98	0.92	0.95	120
252-6	0.98	0.92	0.95	119
252-8	0.98	0.92	0.95	132
253	1.00	0.01	0.02	92
253-2	1.00	0.97	0.98	97
253-4	1.00	0.93	0.96	95
253-6	1.00	0.87	0.93	96
253-8	1.00	0.72	0.84	108
254	1.00	0.27	0.43	55
254-2	1.00	0.82	0.90	59
254-4	1.00	0.70	0.82	59
254-6	1.00	0.61	0.76	58
254-8	1.00	0.42	0.59	68
257	1.00	0.85	0.92	55
257-2	1.00	0.97	0.98	60
257-4	1.00	1.00	1.00	59
257-6	1.00	1.00	1.00	59
257-8	0.91	0.91	0.91	57
258	1.00	0.09	0.17	109
258-2	1.00	0.92	0.96	107
258-4	0.97	0.81	0.88	121
258-6	0.97	0.70	0.81	116
258-8	1.00	0.56	0.72	124
259	0.86	0.06	0.11	96
259-2	0.98	0.92	0.95	108
259-4	0.98	0.92	0.95	120
259-6	0.98	0.92	0.95	107
259-8	0.98	0.92	0.95	105
260	0.92	0.41	0.57	82
260-2	0.96	0.90	0.93	63
260-4	0.96	0.79	0.87	78
260-6	0.95	0.69	0.80	66
260-8	0.94	0.59	0.72	82
261	0.92	0.33	0.49	67
261-2	0.97	0.88	0.92	68
261-4	0.97	0.88	0.92	68
261-6	0.97	0.88	0.92	80
261-8	0.97	0.88	0.92	67
262	0.00	0.00	NaN	54
262-2	1.00	0.79	0.88	53
262-4	1.00	0.61	0.76	56
262-6	1.00	0.42	0.59	53
262-8	1.00	0.21	0.35	66
265	0.80	0.14	0.24	54
266	0.50	0.06	0.11	57
301	0.86	0.75	0.80	95
302	0.93	0.58	0.71	92
303	0.77	0.77	0.77	117
304	0.95	0.96	0.95	93

Instance Matching: IIMB Benchmarks

Data	**Prec**	**Rec**	**F-Meas.**	**Time (sec)**
001	1.00	1.00	1.00	94
002	1.00	1.00	1.00	103
003	1.00	1.00	1.00	125
004	1.00	1.00	1.00	83
005	1.00	0.95	0.97	99
006	1.00	1.00	1.00	105
007	1.00	1.00	1.00	157
008	1.00	0.99	0.99	64
009	1.00	1.00	1.00	97
010	1.00	0.94	0.97	96
011	0.82	0.62	0.71	68
012	1.00	0.96	0.98	91
013	1.00	0.99	0.99	45
014	0.89	0.66	0.76	36
015	0.99	0.95	0.97	65
016	0.93	0.80	0.86	46
017	0.67	0.40	0.50	27
018	0.77	0.54	0.63	51
019	0.88	0.55	0.68	26
020	1.00	1.00	1.00	93
021	1.00	1.00	1.00	93
022	1.00	1.00	1.00	93
023	1.00	1.00	1.00	93
024	1.00	1.00	1.00	93
025	1.00	1.00	1.00	93
026	1.00	1.00	1.00	93
027	1.00	1.00	1.00	93
028	0.46	1.00	0.63	93
029	1.00	1.00	1.00	93
030	0.82	0.57	0.67	65
031	0.83	0.60	0.70	26
032	1.00	0.95	0.97	99
033	1.00	0.93	0.96	95
034	1.00	0.98	0.99	76
035	0.93	0.69	0.79	36
036	0.99	0.86	0.92	95
037	0.83	0.44	0.58	30

Using AgreementMaker to Align Ontologies for OAEI 2009: Overview, Results, and Outlook*

Isabel F. Cruz, Flavio Palandri Antonelli, Cosmin Stroe,
Ulas C. Keles, and Angela Maduko

ADVIS Lab
Department of Computer Science
University of Illinois at Chicago
{ifc|flav|cstroe1|ukeles|maduko}@cs.uic.edu

Abstract. This paper describes our participation in the *Ontology Alignment Evaluation Initiative* (OAEI) 2009 with the AgreementMaker system for ontology matching, in which we obtained excellent results. In particular, we participated in the benchmarks, anatomy, and conference tracks. In the anatomy track, we competed against nine other systems in all four subtracks obtaining the best result in subtrack 3 and the second best result in subtracks 1 and 2. We were also first in finding the highest number of non-trivial correspondences. Furthermore, AgreementMaker came in first place among seven participants in the conference track and achieved the highest precision among all thirteeen participating systems in the benchmarks track. In addition to presenting this year's results, we give an overview of the AgreementMaker system, discuss ways in which we plan to further improve it in the future, and present suggestions for future editions of the OAEI competition.

1 Presentation of the system

As the Semantic Web evolves, more and more ontologies are being developed to describe conceptually several domains of interest. Ontology *matching* or *alignment*, which involves the task of finding correspondences called *mappings* between semantically related entities in two different ontologies, is needed to realize semantic interoperation and heterogenous data integration. A *matching* is a set of mappings established between two ontologies: the *source ontology* and the *target ontology*.

Automatic matching methods are highly desirable to allow for scalability both in the size and number of ontologies being aligned. Our collaboration with domain experts in the geospatial domain [7] has revealed that they value automatic matching methods, especially for ontologies with thousands of concepts. However, they want to be able to evaluate the matching process, thus requiring to be directly involved in the loop. Driven by these requirements, we have developed the AgreementMaker system[1] that integrates efficient automatic matching

* Research supported by NSF Awards ITR IIS-0326284, IIS-0513553, and IIS-0812258.
[1] www.AgreementMaker.org

strategies with a multi-purpose user interface and a module to evaluate matchings [3].

The problem of finding matchings is challenging on several counts. For example, a particular matching method may be effective for a given scenario, but not for others. Also, within the same scenario, the use of different parameters can change the outcome significantly. Therefore, our framework introduces a combined approach that takes advantage of several matching techniques focusing on different features of the ontologies and that allows for different parameters to be set. In particular, our architecture allows for serial and parallel composition where the output of one or more methods can be used as input to another method or several methods can be used on the same input and then combined. A set of mappings may therefore be the result of a sequence of steps called *layers*. The motivation behind this framework is to provide the capability of combining as many mapping layers as needed in order to capture a wide range of relationships between concepts in real-world scenarios [1]. There are parameters that can be defined for all methods, such as cardinality and threshold, whereas other parameters are method dependent. The parameter values can be set manually by the user or by automatic methods that take into account quality measures [2].

We have been developing AgreementMaker since 2001, with a focus on real-world applications [5, 8] and in particular on geospatial applications [4, 6, 7, 9–12, 16]. However, the current version of AgreementMaker and its implementation represents a whole new effort. Not only have we added significant new aspects to the system, but we also have almost completely reimplemented it in the last year. For example, in September of 2008 the previous implementation consisted of 9,000 lines of Java code, whereas in September of 2009 the new implementation had 29,000 lines.

The new AgreementMaker system [1–3] supports: (1) user requirements, as expressed by domain experts; (2) a wide range of input (ontology) and output (agreement file) formats; (3) a large choice of matching methods depending, on the different granularity of the set of components being matched (local vs. global), on different features considered in the comparison (conceptual vs. structural), on the amount of intervention that they require from users (manual vs. automatic), on usage (standalone vs. composed), and on the types of components to consider (schema only or schema and instances); (4) improved performance, that is, accuracy (precision, recall, F-measure) and efficiency (execution time) for the automatic methods; (5) an extensible architecture to incorporate new methods easily and to tune their performance; (6) the capability to evaluate, compare, and combine different strategies and matching results; (7) a comprehensive user interface that supports advanced visualization techniques and a control panel that drives all the matching methods and evaluation strategies; (8) a feedback loop that accepts suggestions and corrections by users and extrapolates new mappings.

1.1 State, purpose, general statement

AgreementMaker comprises a wide range of automatic matching algorithms called *matchers*, an extensible and modular architecture, a multi-purpose user inter-

face, a set of evaluation strategies, and various manual (e.g., visual comparison) and semi-automatic features (e.g., user feedback loop). Given the automatic processing requirement imposed by OAEI, we could mainly make use of the first two features. In particular, we adopted seven different matchers for the competition and took advantage of the modular architecture to organize those matchers into four different matching layers. The evaluation techniques came into play only in the combination phase, to disambiguate the quality of the mappings to be selected.

Even though we could not take direct advantage of the user interface of AgreementMaker in the competition, we want to highlight its benefits prior to the competition. For example, the user interface can display any ontology (the largest ones we have tested have 30,000 concepts), therefore we were able to display the OAEI ontologies to investigate their characteristics (see Figure 1). In addition, we could test, tune, and evaluate both the individual matchers and the particular composition of matchers that we used in the competition.

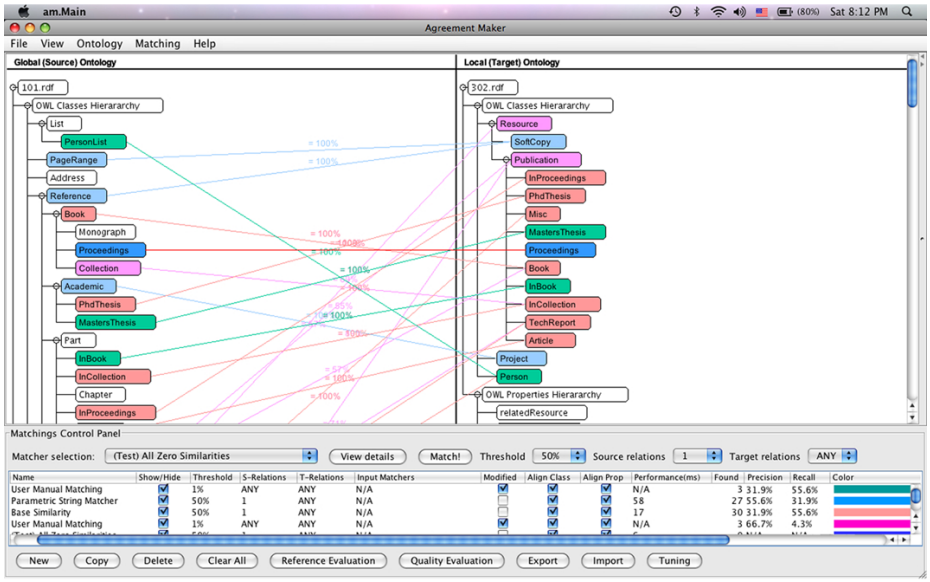


Fig. 1. Graphical User Interface of the AgreementMaker displaying ontologies from the benchmarks track.

1.2 Specific techniques used

For the OAEI 2009 competition, we have created a *stack of matchers*, shown in Figure 2, which are run on the input ontologies to compute the final alignment set.

First, three string-based techniques are independently run on the input ontologies: the *Base Similarity Matcher* (BSM) [7], the *Parametric String-based Matcher* (PSM) [2], and the *Vector-based Multi-word Matcher* (VMM) [2].

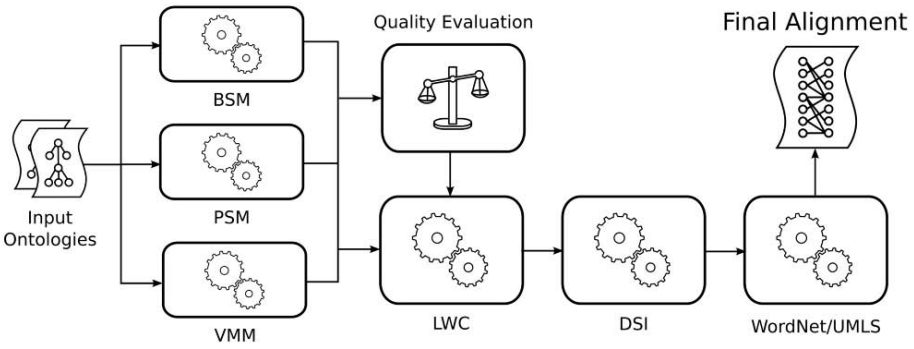


Fig. 2. AgreementMaker OAEI 2009 matcher stack.

BSM is a fundamental string-based matcher, which uses rule-based word stemming, stop word removal, and word normalization in order to find mappings. Going beyond the capabilities of BSM, PSM combines an edit distance measure and a substring measure in order to find mappings. Specifically for this campaign, PSM uses the following formula:

$$\sigma(a,b) = 0.6 * substring(a,b) + 0.4 * edit\_distance(a,b)$$

Our last string similarity matcher, VMM, compiles a *virtual document* for every concept of an ontology, then transforms the strings into TF-IDF vectors and computes the similarity using the *cosine similarity* measure.

After running the string matchers in parallel, their results are combined using the *Linear Weighted Combination* (LWC) matcher [2]. The LWC matcher uses the formula:

$$\sigma_{LWC}(a,b) = w_{BSM} * \sigma_{BSM}(a,b) + w_{PSM} * \sigma_{PSM}(a,b) + w_{VMM} * \sigma_{VMM}(a,b)$$

where the weights for each similarity are automatically calculated using the *local-confidence* quality measure. After the LWC matcher runs, we have a single, combined set of alignments that includes the best alignments from each of the string-based methods. The next matcher, the *Descendant's Similarity Inheritance* (DSI) [7] matcher, is a structure-based matcher that considers the ancestors of the concepts in a mapping in order to increase the similarity of the mapping. The DSI matcher is based on the following heuristic: if two nodes are matched with high similarity, then the similarity between the descendants of those nodes should increase. New mappings are created by the DSI matcher when the similarity of a mapping is increased beyond the threshold established for that matcher. The last step uses a lexical matcher, which considers not only the terms in an ontology, but also the synonyms of those terms as provided by a thesaurus (e.g., WordNet or UMLS).

In order to take advantage of the unique nature of the conference track, we performed an extra computation step, which we used in a new configuration of AgreementMaker called AgreementMakerExt. The OAEI 2009 matcher stack described above considers only two ontologies at a time. In order to expand this consideration, we have added a step that tries to take advantage of the transitivity between ontology mappings. We call this computation the *conflict resolution* step.

As shown in Figure 3, we consider two ontologies O_A and O_B, which have a mapping between them denoted $m_{A \leftrightarrow B}(a_i \in O_A, b_j \in O_B)$, given that concept $a_i \in O_A$ has been matched to concept $b_j \in O_B$. We then consider a third ontology O_C such that concept $a_i \in O_A$ is mapped to some concept $c_k \in O_C$ by mapping $m_{A \leftrightarrow C}(a_i \in O_A, c_k \in O_C)$. We also identify a mapping $m_{B \leftrightarrow C}(b_j \in O_B, c_h \in O_C)$ if there exists a concept $c_h \in O_C$ that matches $b_j \in O_B$. Note that $m_{A \leftrightarrow C}$ and $m_{B \leftrightarrow C}$ may point to different concepts in O_C (i.e., $k \neq h$).

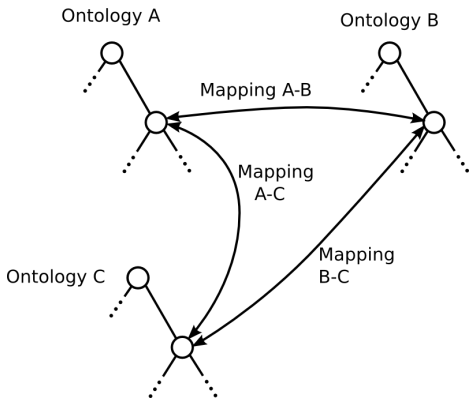


Fig. 3. Conflict resolution using a rating system.

We now implement a rating system. If $m_{A \leftrightarrow C}$ and $m_{B \leftrightarrow C}$ both map to the same concept in O_C (i.e., $k = h$), we increment the rating of all three mappings by 1. If $m_{A \leftrightarrow C}$ or $m_{B \leftrightarrow C}$ does not exist, we decrement the rating of any existing mappings by 1. Likewise, if $m_{A \leftrightarrow C}$ and $m_{B \leftrightarrow C}$ exist, but map to different concepts in O_C (i.e., $k \neq h$), we decrement the rating of all three mappings. This rating is performed for all the mappings between all the ontologies. Finally, we then sweep through the rated mappings and modify the alignments between any two ontologies to choose the mappings that have been rated the highest, resolving any conflicts by choosing the mappings with highest similarity.

1.3 Link to the set of provided alignments (in align format)

AgreementMaker alignment sets for OAEI can be found at
http://www.AgreementMaker.org/OAEI09_Results.zip.

2 Results

In this section, we present the results obtained by AgreementMaker in the OAEI 2009 competition. AgreementMaker participated in three tracks: benchmarks, anatomy and conference. Tests were carried out on a PC running Microsoft Windows Vista 64-bit with Intel Core 2 Duo 2.10 GHz processor and 4 GB RAM.

2.1 Benchmarks

In the benchmarks track, AgreementMaker uses the matchers described in Section 1.2. However, none of the lexical matchers was used in this track. The source ontology is compared to 111 ontologies (including the source ontology) in the same domain (bibliographic references). These ontologies can be grouped into three categories. We describe next the results obtained in each of these three categories as well as the overall results.

Concept test cases (1xx) There are four test cases in this category. AgreementMaker aligned the concept test cases with precision and recall equal to 98%.

Systematic test cases (2xx) For the systematic test cases, AgreementMaker achieved an average precision equal to 98% and an average recall equal to 60%. The average recall is lowered by the results of test cases in which the labels are scrambled. This is due to AgreementMaker's dependence on string mappings in order to find mappings based on structure. AgreementMaker achieved a precision in the range 94% to 100% and a recall in the range 85% to 100% in the systematic test cases in which labels are not scrambled.

Real ontology test cases (3xx) For the four real ontology test cases, AgreementMaker achieved an average precision equal to 92% and an average recall equal to 79%. Precision varied between 83% and 100% while recall varied between 60% and 95%.

Overall The overall results for all the categories place AgreementMaker first with precision equal to 99% and eighth with recall equal to 62% among thirteen participants.

2.2 Anatomy

The anatomy track of OAEI 2009 consists of finding alignments between two large real-world ontologies that are part of Open Biomedical Ontologies (OBO): the adult mouse anatomy (part of the Mouse Gene Expression Database) with 2744 classes and the human anatomy (part of the National Cancer Institute thesaurus) with 3304 classes.

This track consists of four subtracks. AgreementMaker entered all four subtracks using the UMLS Metathesaurus as background knowledge as well as the other modules in the OAEI 2009 matcher stack (see Figure 2).

The reference alignment contains 1523 mappings. Of these mappings only the 988 mappings that form part of the partial reference alignment for subtrack 4 are known. We note, however, that most of those mappings (934) are the "trivial" mappings that can be found by simple string comparison techniques. Therefore, the most important challenge is in finding the non-trivial mappings. The OAEI 2009 competition has released recall values for the non-trivial mappings for subtracks 1 and 3, and named this measure *recall+*. We describe next our results in the four subtracks.

Subtrack 1 In this subtrack, participants are asked to *maximize F-measure*. AgreementMaker used a threshold equal to 0.60 and obtained an F-measure equal to 83.1%, ranking second among ten participants and just short of the first ranked system (SOBOM) with F-measure equal to 85.5% and with a wider distance to the third ranked system (RiMOM) with F-measure equal to 79.3%. AgreementMaker obtained precision equal to 86.5% and recall equal to 79.8%, which was the highest recall value of all ten participants. AgreementMaker also ranked first in recall+, which was equal to 48.9%. The runtime was 23 minutes.

Subtrack 2 In this subtrack, participants are asked to *maximize precision*. AgreementMaker used a threshold equal to 0.75 and obtained precision equal to 96.7%, ranking second among seven participants and just short of the first ranked system (DSSim) with precision equal to 97.3% and with a wider distance to the third ranked system (TaxoMap) with precision equal to 95.3%. The runtime was 25 minutes.

Subtrack 3 In this subtrack, participants are asked to *maximize recall*. AgreementMaker used a threshold equal to 0.35. This choice of threshold combined with the UMLS module that was used to provide background knowledge resulted in our system having the highest recall among seven participants. AgreementMaker achieved recall equal to 81.5%. The second ranked system (Lily) had recall equal to 77.4%. AgreementMaker also ranked first in recall+, which was equal to 55.3%. The runtime was 17 minutes.

Subtrack 4 In this subtrack, participants are asked to *maximize precision, recall, and F-measure* using the mappings in a *partial reference alignment*, which is provided. AgreementMaker obtained the highest increase in precision among the four participants in this subtrack (+12.8%), significantly higher than that of the second ranked participant (ASMOV, with +3.4%). However, for recall and F-measure it was last (-18.1% and -6.3%, respectively).

2.3 Conference

In this track, participants are asked to find all correct correspondences (equivalence and/or subsumption correspondences) in a collection of 15 ontologies that describe a domain associated with the organization of conferences. Participants need to compute the set of mappings for each pair of ontologies. We note that for two ontologies O_i and O_j, $i \neq j$, once matching $M(O_i, O_j)$ is computed, the

symmetric matching $M(O_j, O_i)$ need not be computed. Therefore, 105 alignment files need to be computed. In our case, the alignment files contained 2070 individual alignments in total.

We entered the competition with two different strategies. In one of them we used the OAEI 2009 matcher stack (see Figure 2), while in the other one, which we call AgreementMakerExt, we performed an extra computation step, which we described in Section 1.2. This step allows for more than two ontologies to be considered at a time by taking advantage of transitivity among ontology mappings. We call this computation the *conflict resolution* step.

From the "evaluation based on reference alignment" we see that AgreementMaker did very well overall. The alignments obtained by AgreementMaker with a threshold equal to 0.75 were the best among the seven participating systems, with precision equal to 69%, recall equal to 51%, and F-measure equal to 57%. The results obtained by AgreementMakerExt were also good, but the conflict resolution step reduced precision (to 54%), which led to a reduction of F-measure (to 51%). Since our system does not produce subsumption relations, it could not be evaluated on "restricted semantic precision and recall". Finally, in the "evaluation based on manual labeling", which rates how well the certainty of a system correlates with the correctness of the mappings, $80\% \pm 6\%$ of the mappings that were rated by AgreementMaker with a similarity equal to 1.0 were correct.

2.4 Comments on the obtained results

Benchmarks. Although AgreementMaker achieved the highest precision (99%) among all thirteen participating systems, it was less successful in terms of finding certain kinds of mappings, thus leading to less good recall (62%). This is because our structural techniques depend on lexical mappings that need to be found previously. When there were no lexical similarities, as was the case with some of the systematic test cases (2xx) where textual information was randomly modified, structural similarities were not found.

Anatomy. In subtracks 1-3, the most difficult task consists of finding those mappings that are non-trivial as observed in Section 2.2. Even so, AgreementMaker did very well in these subtracks having achieved first place in subtrack 3, second place in the remaining two, and first place in recall for non-trivial mappings. The key to further improvement relies on new techniques for finding other non-trivial correspondences.

In subtrack 4, as indicated in the work by Lambrix and Liu [14], partial reference alignments can be used at several points in the alignment process. We used the partial reference alignment that was provided in two ways:

1. To partition the ontologies into mappable parts so that every concept in the source ontology is not compared to every concept in the target ontology. We were able to reduce the running time of our algorithms by about 75%.
2. To remove mappings that are considered incorrect. Once the mappable parts are created, we assume that given a mappable part in the source ontology

and its corresponding mappable part in the target ontology, concepts at the same depth in the hierarchy match in the two mappable parts. We observe that this is especially true for ontologies that have similar structure.

Finally, partial reference alignments may be used to add undiscovered mappings to the final alignment results. This third aspect of using the partial reference alignment presents the most difficult challenges. We have not yet been able to implement a satisfactory method for accomplishing this third task. We hope to investigate this problem in future work.

Conference. AgreementMaker did very well on the conference track. AgreementMakerExt also did well in spite of the observed decrease in precision. In fact, the conflict resolution step decreased precision, while keeping recall almost the same. This leads us to infer that the conflict resolution step added wrong mappings and removed some correct mappings. We note, however, that the official results for this track were obtained using a partial reference matching that is one fifth the size of the full reference matching. The conflict resolution step works globally, that is, it may improve results overall, but not necessarily for just a "slice" of the problem; we therefore conjecture that this could provide the justification for the decrease in precision.

As for improving on the obtained results, our system ranked first with precision equal to 69% and recall equal to 51%, considering a threshold equal to 0.75. Precision can be further improved by understanding which mappings were erroneously included in the alignments, thus requiring an investigation of every single mapping in the alignment. Unfortunately, without the reference alignment we can only make an educated guess about which mappings are correct or incorrect. As far as improving recall, it seems that there is semantic information that we are not considering when aligning the ontologies. This may have to do with the unique nature of the conference track, in that it considers 15 ontologies mapped against one another instead of the traditional two.

2.5 Proposed new measures

In the anatomy subtracks 2 and 3, the participants are asked to compute an alignment that maximizes respectively precision (subtrack 2) and recall (subtrack 3). However, results that are based solely on the maximization of precision or recall may not be conclusive. For instance, a system could easily produce an alignment with 100% precision by computing an empty set of mappings, while an alignment containing all possible correspondences would have a 100% recall. Therefore, we suggest a different ranking system based on the use of a properly configured F-measure. To define our proposal, we first consider the definition of F-measure.

Given a set of mappings M and a reference matching R, the *F-measure of M with respect to R* is given by the following expression:

$$F\text{-}measure(M,R) = \frac{precision(M,R) \cdot recall(M,R)}{(1-\alpha) \cdot precision(M,R) + \alpha \cdot recall(M,R)}$$

The higher the value of $\alpha \in [0,1]$, the more important is precision with respect to recall. Generally, it is set to 0.5 to get the harmonic mean of the two measures. In order to rank the matching results of the anatomy subtracks 2 and 3, we propose that they should be measured with respect to F-measure (not precision for subtrack 2 and recall for subtrack 3). Therefore, α should be greater than 0.5 for subtrack 2 and lower than 0.5 for subtrack 3. The value for α could be chosen by considering a ranking among the results obtained for the anatomy subtracks 2 and 3 from previous years. Once that ranking is established, then the corresponding value of α would be given to the OAEI participants in future editions of the competition so that they can tune their methods for that particular value.

Finally, we want to point to the fact that AgreementMaker can be used to import the OAEI alignments computed by any matching system in order to evaluate precision, recall, and F-measure thus allowing for their direct comparison. In addition, AgreementMaker can evaluate structural discrepancy measures [13] and the *local-confidence* quality measure that we defined [2]. We further plan to implement the incoherence-based quality measure [15].

2.6 Discussions on ways to improve the proposed system

There are several directions that we would like to explore to improve AgreementMaker. For example, we want to add matchers that rely solely on the structure of the ontologies to find matchings. The DSI (Descendant's Similarity Inheritance) and SSC (Sibling's Similarity Contribution) structure-based matchers exploit the structure of the ontology by respectively considering the concepts that have as descendants or siblings the concepts being matched [7, 16]. However, they first rely on similarity values computed by string-based matchers. We hope to devise "pure" structure-based matchers that would work in the benchmarks track cases where AgreementMaker did not produce any mappings, even though the ontologies being matched are very similar structurally.

AgreementMaker was not able to fully exploit the unique nature of the conference track. One way to further improve our results in the conference track is to incorporate the capabilty of extending alignments over multiple ontologies, instead of considering only two ontologies at a time.

Finally, we will further explore how to use the provided partial reference alignment in subtrack 4 of the anatomy track. In particular, we encountered some false negatives due to the dissimilarity in structure of the two anatomy track ontologies. We hope to devise other techniques that circumvent this. In addition, we would like to use the partial reference alignment to discover non-trivial mappings.

However, we are not only focusing on automatic ontology matching. We believe that involving the user in the matching process is crucial in finding the mappings that are not found by automatic methods. By taking advantage of the multi-purpose user interface of the AgreementMaker, we have been working on a semi-automatic matching approach that ranks concepts according to their relevance and presents to users the top-k most relevant concepts together with

the most likely mappings associated with them. In addition, our solution encompasses a feedback loop that extrapolates new correspondences and corrects wrong mappings.

3 Conclusions

In this paper we gave an overview of the new AgreementMaker system, which was developed in the last year, presented the results that this system obtained in the OAEI 2009 competition for the benchmarks, anatomy, and conference tracks, and discussed those results. We also proposed new measures for future OAEI competitions.

In the benchmarks track, AgreementMaker found alignments (a total of 111) for all cases. All those alignments were computed in less than 3 minutes with an overall precision equal to 99% (highest among 13 competing systems) and an overall recall equal to 62% (eighth place).

AgreementMaker participated in all four subtracks of the anatomy track, placing second in subtracks 1 and 2 and first in subtrack 3 among ten, seven, and seven participants, respectively. AgreementMaker also found the highest number of non-trivial correspondences. In the last subtrack, subtrack 4, it achieved the highest improvement in precision among four participants together with an improved execution time.

AgreementMaker was also very successful in the conference track: it achieved the best results among seven participants, with precision equal to 69% and recall equal to 51% for a threshold equal to 0.75.

Overall, AgreementMaker exhibited an excellent performance in the OAEI 2009 competition. However, the competition only tests the component of AgreementMaker that performs automatic matchings. The automatic matching capabilities of AgreementMaker are just a small part of a full framework for ontology matching, which also supports the visualization of ontologies and the evaluation of their matchings. Those matchings can also be produced manually, semi-automatically, or using an extrapolating mechanism that accepts input from users. Several of these components of AgreementMaker, even if not directly tested in the competition, were quite useful for "understanding" the ontologies and for the tuning and evaluation of the matching strategies. However, we believe that there is still room for improvement and we plan to continue our quest for efficiency and effectiveness in the ontology matching process.

References

1. Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. *PVLDB*, 2(2):1586–1589, 2009.
2. Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. Efficient Selection of Mappings and Automatic Quality-driven Combination of Matching Methods. In *ISWC International Workshop on Ontology Matching*. CEUR-WS, 2009.

3. Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. Integrated Ontology Matching and Evaluation. In *International Semantic Web Conference (Posters & Demos)*, 2009.
4. Isabel F. Cruz and Afsheen Rajendran. Exploring a New Approach to the Alignment of Ontologies. In *ISWC Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data*, volume 83 of *CEUR-WS*, pages 7–12, 2003.
5. Isabel F. Cruz and Afsheen Rajendran. Semantic Data Integration in Hierarchical Domains. *IEEE Intelligent Systems*, March-April:66–73, 2003.
6. Isabel F. Cruz, Afsheen Rajendran, William Sunna, and Nancy Wiegand. Handling Semantic Heterogeneities Using Declarative Agreements. In *ACM International Symposium on Advances in Geographic Information Systems (ACM GIS)*, pages 168–174, 2002.
7. Isabel F. Cruz and William Sunna. Structural Alignment Methods with Applications to Geospatial Ontologies. *Transactions in GIS, special issue on Semantic Similarity Measurement and Geospatial Applications*, 12(6):683–711, 2008.
8. Isabel F. Cruz, William Sunna, and Anjli Chaudhry. Ontology Alignment for Real-World Applications. In *National Conference on Digital Government Research (dg.o)*, pages 393–394, 2004.
9. Isabel F. Cruz, William Sunna, and Anjli Chaudhry. Semi-Automatic Ontology Alignment for Geospatial Data Integration. In *International Conference on Geographic Information Science (GIScience)*, volume 3234 of *Lecture Notes in Computer Science*, pages 51–66. Springer, 2004.
10. Isabel F. Cruz, William Sunna, Nalin Makar, and Sujan Bathala. A Visual Tool for Ontology Alignment to Enable Geospatial Interoperability. *Journal of Visual Languages and Computing*, 18(3):230–254, 2007.
11. Isabel F. Cruz, William G. Sunna, and Kalyan Ayloo. Concept Level Matching of Geospatial Ontologies. In *GIS Planet International Conference and Exhibition on Geographic Information*, 2005.
12. Isabel F. Cruz and Huiyong Xiao. Data Integration for Querying Geospatial Sources. In John Sample, Kevin Shaw, Shengru Tu, and Mahdi Abdelguerfi, editors, *Geospatial Services and Applications for the Internet*, pages 113–137. Springer, 2008.
13. Cliff Joslyn, Patrick Paulson, and Amanda White. Measuring the Structural Preservation of Semantic Hierarchy Alignment. In *ISWC International Workshop on Ontology Matching*. CEUR-WS, 2009.
14. Patrick Lambrix and Qiang Liu. Using Partial Reference Alignments to Align Ontologies. In *European Semantic Web Conference*, volume 5554 of *Lecture Notes in Computer Science*, pages 188–202, 2009.
15. Christian Meilicke and Heiner Stuckenschmidt. Incoherence as a Basis for Measuring the Quality of Ontology Mappings. In *ISWC International Workshop on Ontology Matching*, volume 431 of *CEUR-WS*, 2008.
16. William Sunna and Isabel F. Cruz. Structure-Based Methods to Enhance Geospatial Ontology Alignment. In *International Conference on GeoSpatial Semantics (GeoS)*, volume 4853 of *Lecture Notes in Computer Science*, pages 82–97. Springer, 2007.

AROMA results for OAEI 2009

Jérôme David[1]

Université Pierre-Mendès-France, Grenoble
Laboratoire d'Informatique de Grenoble
INRIA Rhône-Alpes, Montbonnot Saint-Martin,
France
Jerome.David-at-inrialpes.fr

Abstract. This paper presents the results obtained by AROMA for its second participation to OAEI. AROMA is an hybrid, extensional and asymmetric ontology alignment method that makes use of the association paradigm and a statistical interestingness measure, the implication intensity. AROMA performs a post-processing step that includes a terminological matcher. This year we modify this matcher in order to improve the recall obtained on real-case ontology, i.e. anatomy and 3xx tests.

1 Presentation of AROMA

1.1 State, purpose, general statement

AROMA is an hybrid, extensional and asymmetric matching approach designed to find out relations of equivalence and subsumption between entities, i.e. classes and properties, issued from two textual taxonomies (web directories or OWL ontologies). Our approach makes use of the association rule paradigm [Agrawal *et al.*, 1993], and a statistical interestingness measure. AROMA relies on the following assumption: *An entity A will be more specific than or equivalent to an entity B if the vocabulary (i.e. terms and also data) used to describe A, its descendants, and its instances tends to be included in that of B.*

1.2 Specific techniques used

AROMA is divided into three successive main stages: (1) The pre processing stage represents each entity, i.e. classes and properties, by a set of terms, (2) the second stage consists of the discovery of association rules between entities, and finally (3) the post processing stage aims at cleaning and enhancing the resulting alignment.

The first stage constructs a set of relevant terms and/or datavalues for each class and property. To do this, we extract the vocabulary of class and property from their annotations and individual values with the help of single and binary term extractor applied to stemmed text. In order to keep a morphism between the partial orders of class and property subsumption hierarchies in one hand and the inclusion of sets of term in the other hand, the terms associated with a class or a property are also associated with its ancestors.

The second stage of AROMA discovers the subsumption relations by using the association rule model and the implication intensity measure [Gras *et al.*, 2008]. In the context of AROMA, an association rule $a \rightarrow b$ represents a quasi-implication (i.e. an implication allowing some counter-examples) from the vocabulary of entity a into the vocabulary of the entity b. Such a rule could be interpreted as a subsumption relation from the antecedent entity toward the consequent one. For example, the binary rule $car \rightarrow vehicle$ means: "The concept car is more specific than the concept $vehicle$". The rule extraction algorithm takes advantage of the partial order structure provided by the subsumption relation, and a property of the implication intensity for pruning the search space.

The last stage concerns the post processing of the association rules set. It performs the following tasks:

- deduction of equivalence relations,
- suppression of cycles in the alignment graph,
- suppression of redundant correspondences,
- selection of the best correspondence for each entity (the alignment is an injective function),
- the enhancement of the alignment by using a string similarity -based matcher and previously discovered correspondences.

This year, we made some changes on the string similarity -based matcher. These changes are primarily designed to improve the recall on anatomy track. Now AROMA includes an equality -based matcher: two entities are considered equivalent if they share at least one annotation. This matcher is only applied on unaligned pairs of entities.

The string similarity based matcher still makes use of Jaro-Winkler similarity but relies on a different weighting scheme. As an ontology entity is associated to a set of annotations, i.e. local name, labels and comments, we need a collection measure for aggregating the similarity values between all entity pairs. Last year, we relied on maximal weight maximal graph matching collection measure, see [David and Euzenat, 2008] for details.

In order to favour the measure values of most similar annotations pairs, we choose to use the following collection measure:

$$\Delta_{mw}(e, e') = \begin{cases} \frac{\sum_{a \in T(e)} \arg\max_{a' \in T(e')} sim_{jw}(a,a')^2}{\sum_{a \in T(e)} \arg\max_{a' \in T(e')} sim_{jw}(a,a')} & \text{if } |T(e)| \leq |T(e')| \\ \Delta_{mw}(e', e) & \text{otherwise} \end{cases}$$

where $T(e)$ is the set which contains the annotations and the local name of e, and sim_{jw} is the Jaro-Winkler similarity. For all OAEI tracks, we choose a threshold value of 0.8.

For more details about AROMA, the reader should refer to [David *et al.*, 2007; David, 2007].

1.3 Link to the system and parameters file

The version 1.1 of AROMA has been used for OAEI2009. This version can be downloaded at : `http://gforge.inria.fr/frs/download.php/23649/AROMA-1.1.zip`.

The command line used for aligning two ontologies is:

```
java -jar aroma.jar onto1.owl onto2.owl [alignfile.rdf]
```

The resulting alignment is provided in the alignment format.

1.4 Link to the set of provided alignments (in align format)

```
http://www.inrialpes.fr/exmo/people/jdavid/oaei2009/results_AROMA_oaei2009.zip
```

2 Results

We participated to the benchmark, anatomy and conference tracks. We used the same configuration of AROMA for all tracks. We did not experience scaling problem. Since AROMA relies on syntactical data without using any multilingual resources, it is not able to find alignment on the multilingual library track. Finally, we also did not participate either to the instance matching track since AROMA is not designed for such a task.

2.1 Benchmark

Since AROMA mainly relies on textual information, it obtains bad recall values when the alterations affect all text annotations both in the class/property descriptions and in their individual/property values. AROMA does not seem to be influenced by structural alterations (222-247). On these tests, AROMA favours high precision values in comparison to recall values.

In comparison with last year, the modification made on AROMA have a limited negative impact on 2xx tests. By contrast, the results on 3xx tests have been enhanced: from 82% of precision and 71% of recall to respectively 85% and 78%.

2.2 Anatomy

On anatomy test, we do not use any particular knowledge about biomedical domain. AROMA runs quite fast since it takes benefits of the subsumption relation for pruning the search space. We further optimized the code since last year and now AROMA needs around 1 min. to compute the alignment. This pruning feature used by AROMA partially explained the low recall values obtained last year. For this edition, we enhanced the recall by using also an string equality based matcher before using the lexical similarity based matcher. Since AROMA returns not only equivalence correspondences but also subsumption correspondences, its precision value is negatively influenced. It could be interesting to evaluate results by using semantic precision and recall.

3 General comments

3.1 Comments on the OAEI test cases

In this section, we give some comments on the directory and oriented matching tracks of OAEI.

Directory The two large directories, that were given in previous editions of OAEI, are divided into very small sub directories. AROMA cannot align such very small directories because our method is based on a statistical measure and then it needs some large amount of textual data. However, AROMA discovers correspondences when it is applied to the complete directories. It would be interesting to reintroduce these large taxonomies for the next editions.

Oriented matching We did not participate to this track because we think that it is not well designed. Indeed, the proposed reference alignments are not complete.

For example in the 303 test, the reference alignment contains:

- 101#MastersThesis $\leq$ 103#Academic
- 103#MastersThesis $\leq$ 101#Academic

Obviously, no reliable matching algorithm would return these two correspondences but rather:

- 101#MastersThesis $\equiv$103#MastersThesis
- 101#Academic $\equiv$ 103#Academic

In addition, from these two last correspondences, we could easily deduce the two first ones.

Our suggestion for designing a better oriented matching track would be to remove some classes and properties in the target ontologies so as to obtain complete reference alignments with some subsumption relations. For example, it would be more accurate to remove the concept MasterThesis from the ontology 103 in order to naturally change 101#MastersThesis $\equiv$103#MastersThesis by 101#MastersThesis $\leq$ 103#Academic in the reference alignment.

4 Conclusion

The version of AROMA includes a new matcher based on annotation equality. This change allows better time efficiency because it reduces the number of unaligned entities before the use of a more time consuming terminological matcher. Furthermore, we obtained better results on the 3xx tests of benchmark and tend to enhance the recall obtained on anatomy track.

References

[Agrawal *et al.*, 1993] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216. ACM Press, 1993.

[David and Euzenat, 2008] Jérôme David and Jérôme Euzenat. Comparison between ontology distances (preliminary results). In *Proceedings of the 7th International Semantic Web Conference*, volume 5318 of *Lecture Notes in Computer Science*, pages 245–260. Springer, 2008.

[David *et al.*, 2007] Jérôme David, Fabrice Guillet, and Henri Briand. Association rule ontology matching approach. *International Journal on Semantic Web and Information Systems*, 3(2):27–49, 2007.

[David, 2007] Jérôme David. *AROMA : une méthode pour la découverte d'alignements orientés entre ontologies à partir de règles d'association*. PhD thesis, Université de Nantes, 2007.

[Gras *et al.*, 2008] Régis Gras, Einoshin Suzuki, Fabrice Guillet, and Filippo Spagnolo, editors. *Statistical Implicative Analysis, Theory and Applications*, volume 127 of *Studies in Computational Intelligence*. Springer, 2008.

ASMOV: Results for OAEI 2009

Yves R. Jean-Mary[1], E. Patrick Shironoshita[1], Mansur R. Kabuka [1,2]

[1] INFOTECH Soft, 1201 Brickell Ave., Suite 220, Miami, Florida, USA 33131
[2] University of Miami, Coral Gables, Florida, USA 33124
{reggie, kabuka}@infotechsoft.com

Abstract. The Automated Semantic Mapping of Ontologies with Validation (ASMOV) algorithm for ontology alignment was one of the top performing algorithms in the 2007 and 2008 Ontology Alignment Evaluation Initiative (OAEI) contests. In this paper, we present a brief overview of the algorithm and its improvements, followed by an analysis of its results on the 2009 OAEI tests.

1 Presentation of the System

In recent years, ontology alignment has become popular in solving interoperability issues across heterogonous systems in the semantic web. Though many techniques have emerged from the literature [1], the distinction between them is accentuated by the manner in which they exploit the ontology features. ASMOV, an algorithm that automates the ontology alignment process, uses a weighted average of measurements of similarity along four different features of ontologies, obtains a pre-alignment based on these measurements, and then semantically verifies this alignment to ensure that it does not contain semantic inconsistencies. A more complete description of ASMOV is presented in [3].

1.1 State, Purpose, General Statement

ASMOV is an automatic ontology matching tool which has been designed in order to facilitate the integration of heterogeneous data sources modeled as ontologies. The current ASMOV implementation produces mappings between concepts, properties, and individuals, including mappings between object and datatype properties.

1.2 Specific Techniques Used

The ASMOV algorithm iteratively calculates the similarity between entities for a pair of ontologies by analyzing four features: lexical elements (id, label, and comments), relational structure (ancestor-descendant hierarchy), internal structure (property restrictions for concepts; types, domains, and ranges for properties; data values for individuals), and extension (instances of classes and property values). The measures obtained by comparing these four features are combined into a single value using a

weighted sum in a similar manner to [2]. These weights have been optimized based on the OAEI 2008 benchmark test results.

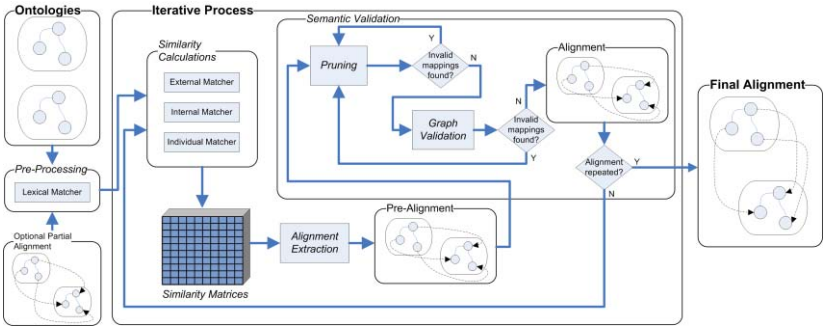


Fig. 1. The ASMOV Mapping Process

Fig. 1 illustrates the fully automated ASMOV mapping process, which has been implemented in Java. In the pre-processing phase, the ontologies are loaded into memory using the Jena ARP parser [4] and ASMOV's ontology modeling component. A thesaurus is then used in order to calculate the lexical similarities between each pair of concepts, properties and individuals. ASMOV can be configured to use either the UMLS Metathesaurus [5] or WordNet [6] in order to derive the similarity measures. A user can also opt to not use a thesaurus; in that case, a text matching algorithm is used to compute the lexical distance.

Following this, the similarities between pairs of entities along the relational structure, internal structure, and extensional dimensions are calculated, and an overall similarity measure (or confidence value) is stored in three two-dimensional matrices, one each for concepts, properties, and individuals. From these similarity matrices, a pre-alignment is obtained by selecting the entity from one ontology with the highest confidence value for a corresponding entity in the other ontology.

This pre-alignment then goes through semantic verification, which detects semantically inconsistent mappings and their causes. These inconsistent mappings are removed from the pre-alignment and logged so that the algorithm does not attempt to map the same entities in a subsequent iteration; mappings are removed from the log of inconsistencies when the underlying cause disappears. Five specific types of inconsistencies are detected by ASMOV:

- Multiple entity correspondences, where the same entity on one ontology is mapped with multiple entities in the other ontology; unless these multiple entities are asserted to be equivalent, this type of mapping is unverified.
- Crisscross correspondences, where if a class c_1 in one ontology is mapped to some other class c_1' in the second ontology, a child of c_1 cannot be mapped to a parent of c_1'.
- Disjointness-subsumption contradiction, where if two classes c_1 and c_2 are disjoint in one ontology, they cannot be mapped to two other classes c_1' and c_2' in the second ontology where one is subsumed by the other. This also

applies to the special cases where c_1' and c_2' are asserted equivalent, or where they are identical.

- Subsumption incompleteness, if two classes c_1 and c_2 are mapped to two other classes c_1' and c_2' respectively in the second ontology, and if c_2 is subsumed by c_1, then c_2' must be subsumed by c_1', otherwise the correspondences are unverified. Similar incompleteness can be verified for the special case of equivalence.
- Domain and range incompleteness: if a class c_1 in one ontology is mapped to some class c_1' in the second ontology, and a property p_1 in the first ontology is mapped to some property p_1' in the second ontology, and if c_1 belongs to the domain (or range) of p_1 , then c_1' must belong to the domain (or, equivalently, range) of p_1',

Since OAEI 2008, ASMOV has been improved in three important respects. In particular, instance matching, which had been initially implemented in previous versions, has been thoroughly redesigned, due to the availability of high-quality reference alignments for testing. As can be seen in the results section, this has resulted in high accuracy for the matching of instances, and has also had an effect in the improvement of the accuracy of class and property matching. In addition, the code base for the entire implementation of ASMOV has been thoroughly debugged and tested, particularly to ensure faithful derivation of the entity-set similarity calculations and the semantic verification process as described in [3]. Further, for the anatomy tests in particular, we have worked to improve the performance of the UMLS Metathesaurus adapter, resulting in a significant improvement in execution time.

1.3 Adaptations Made for the Evaluation

No special adaptations have been made to the ASMOV system in order to run the 2009 OAEI tests; however, five Java executable classes have been added in order to respectively run the benchmark series of tests, the anatomy tests, the directory tests, the FAO tests, and the conference tests, and output the results in the OAEI alignment format. The threshold function used to determine the stop criteria for ASMOV was established as a step function, 95% for alignments where both ontologies have more than 500 concepts, and 100% otherwise. Although the rules of the contests stated that all alignments should be run from the same set of parameters, it was necessary to change two parameters for the anatomy tests. These parameters relate to the thesaurus being used (UMLS instead of WordNet) and to the flag indicating whether or not to use ids of entities in the lexical similarity calculations.

1.4 Link to the ASMOV System

The ASMOV system (including the parameters file) can be downloaded from http://support.infotechsoft.com/integration/ASMOV/OAEI-2009.

1.5 Link to the Set of Alignments Produced by ASMOV

The results of the 2008 OAEI campaign for the ASMOV system can be found at http://support.infotechsoft.com/integration/ASMOV/OAEI-2009.

2 Results

In this section, we present our comments on the results obtained from the participation of ASMOV in the five tracks of the 2009 Ontology Alignment Evaluation Initiative campaign. All tests were carried out on a PC running FreeBSD over VMware with two quad-core Intel Xeon processor (1.86 GHz), 8 GB of memory, and 2x4MB cache. Since the tests in the 2008 version were run in a similar machine, but running SUSE Linux Enterprise Server 10 directly on the processors, the execution times are not directly comparable, and should only be used as guidelines.

2.1 Benchmark

The OAEI 2009 benchmark tests have been divided by the organizing committee in eleven levels of difficulty; we have added one more level to include the set of 3xx tests, which have been included in the benchmark for compatibility with previous years. In Table 1, we present the results of running our current implementation of ASMOV against the OAEI 2009 tests, in comparison with the results obtained in the tests in 2008 [7], where ASMOV was found to be one of the top three performing systems [8]. As can be seen, the precision, recall, and F1 measure for the entire suite of tests shows the current implementation of ASMOV achieves 95% precision and 87% recall, and an F1 measure of 91%, which represents a 1% improvement over the 2008 version. The total execution time for all tests was 161 sec..

The accuracy of ASMOV in the benchmark tests is very high, especially for the

Table 1. Benchmark test results for ASMOV version 2009 and version 2008

Level	ASMOV 2009			ASMOV 2008		
	Precision	Recall	F1	Precision	Recall	F1
0	1.00	1.00	1.00	1.00	1.00	1.00
1	1.00	1.00	1.00	1.00	1.00	1.00
2	1.00	0.99	0.99	1.00	0.99	0.99
3	0.99	0.98	0.98	0.98	0.97	0.97
4	0.99	0.98	0.98	0.99	0.98	0.98
5	0.97	0.93	0.95	0.96	0.93	0.94
6	0.95	0.88	0.91	0.94	0.88	0.91
7	0.94	0.83	0.88	0.93	0.83	0.88
8	0.91	0.71	0.80	0.90	0.71	0.79
9	0.83	0.48	0.61	0.78	0.46	0.58
10	0.40	0.04	0.07	0.40	0.04	0.07
3xx	0.81	0.82	0.81	0.81	0.77	0.79
All	**0.95**	**0.87**	**0.91**	**0.95**	**0.86**	**0.90**

lowest levels of difficulty. It is particularly noteworthy that improvements in both precision and recall were obtained especially at higher levels, with the largest improvement within level 9, the second most difficult. We attribute these improvements mostly to the standardization of the entity-set similarity calculation, as well as to the correction of some coding errors. There is also significant improvement, especially in recall, in testing with the real-world ontologies.

2.2 Anatomy

For the anatomy track, ASMOV uses the UMLS Metathesaurus [5] instead of WordNet in order to more accurately compute the lexical distance between medical concepts. Importantly, improvement in execution time of more than one order of magnitude, for all tests, was achieved by pre-indexing the UMLS Metathesaurus. In addition, the lexical similarity calculation between concept names (ids) is ignored as instructed by the track organizers. ASMOV produces an alignment for all four subtasks of this track:

1. *Optimal solution*: The optimal solution alignment is obtained by using the default parameter settings of ASMOV. It finds 1235 correct and 49 incorrect mappings from the partial alignment, Its total execution time was 4.1 minutes, an order of magnitude improvement over 2008, when it took almost 4 hours.
2. *Optimal precision*: The alignment with optimal precision is obtained by changing the threshold for valid mappings from 0% to 30%. This means that only mappings with confidences greater or equal to 0.3 make it to the alignment. This finds 1,187 correct and only 30 incorrect mappings.from the partial alignment The time cost for the generation of this alignment was 2.7 minutes, compared to 3 hours and 50 minutes in 2008.
3. *Optimal recall*: The alignment with optimal recall is generated by using a threshold of 0% and turning off the semantic verification process, to allow more mappings to form part of the final alignment. Under this setup, 1278 correct mappings and 55 incorrect mappings from the partial alignment are found. it took 4.4 minutes to execute, as opposed the 2008 execution time of 5 hours and 54 minutes.
4. *Extended solution*: The alignment, using the same setup as the optimal solution but with the partial alignment given as input, was obtained in 2.51 min.

2.3 Conference

This collection of tests dealing with conference organization contains 15 ontologies. ASMOV is able to generate all 105 potential alignments from those ontologies, as opposed to 2008 when only 75 alignments were processed. The overall time required to process all 105 correspondences was 187 seconds.

2.4 Directory

The directory tests were completed in 2.75 minutes for all 4639 tasks involved; this is in the same order of magnitude as the 2008 version. A manual analysis of a small sample of the correspondences found in these tests shows that a number of possibly erroneous correspondences found by ASMOV have a very low, but non-zero, confidence value. We therefore expect that the reported accuracy of ASMOV will suffer as a result. Note that the tests were run without setting a confidence value threshold, in compliance with the indication that all tracks be run with the same set of parameters; the use of a threshold would eliminate many of these potentially erroneous correspondences.

2.5 Oriented matching

We have performed gold-standard based evaluation on the Artificial Ontologies Corpus, derived from the benchmark series corpus of 2006 (a subset of the current benchmark series); due to time constraints, it was not possible to obtain results for the Real World Ontologies Corpus, In the Artificial Ontologies Corpus, ASMOV obtains an overall accuracy of 90% precision and 89% recall; in several of the simpler tests, ASMOV finds 100% precision and recall; some reduction in accuracy is observed for the more difficult tests. The execution time for these tests was 75.7 sec.

2.6 Instance Matching

Previous versions of ASMOV contained mechanisms for instance matching based on the same principles as for class and property matching, as outlined in [3]. However, the lack of a gold standard had precluded us from performing rigorous testing on these procedures. With the availability of the instance matching track in OAEI 2009, we have been able to test and improve our algorithms.

The performance of ASMOV in the set of IIMB instance matching tests is quite good, achieving an overall precision very close to 100% and overall recall of 98%. Perfect results are obtained for all the value transformation tests (002-010), as well as for the logical transformation tests (020-029). For the structural transformation tests, slight reductions in accuracy, especially in recall, are found for tests 015, 016, 017, and 019. This slight decrease results from conditions where the best match for an instance in one ontology cannot be chosen from among two or more alternatives in the other ontology; in these cases, ASMOV prefers to not make a selection. The same condition affects the result of test 031.The execution time for all 37 tests was 28 min. 15 sec.; the comparatively longer time is due to the large number of entities in each Abox.

Of the remainder of the instance matching tests, memory consumption issues prevented us from running most of the tests. The only test that could be run was to align the ePrints and Rexa ontologies; this test took 5.7 secs., to execute. The results from this test, and a manual analysis of some results, show that it is necessary to

improve some aspects of instance matching, such as the matching of names where either the first or last name is inserted first in the label of an instance. This also shows that it is necessary to improve the scalability of ASMOV when very large ontologies are being aligned.

3 General Comments

3.1 Comments on the Results

The current version of ASMOV has shown improvement overall in recall and F1 measure with respect to the results obtained last year. This is significant since the results in 2008 were already very high. Particularly important is the fact that the improvements have been obtained within some of the most difficult tests, showing the versatility of ASMOV in finding alignments under various conditions. The high accuracy results from the IIMB instance matching tests also show the capability of ASMOV in these tasks. In the anatomy tests, an improvement in execution time of more than one order of magnitude was obtained, and we also expect that the accuracy of the results should have increased with respect to 2008.

3.2 Discussions on the Way to Improve ASMOV

ASMOV still needs to improve its ability to work with very large ontologies and resources. The current implementation of the algorithm utilizes a memory-based approach, where the entire contents of the ontologies are loaded in memory prior to performing the matching process. This process needs to be modified to use permanent storage in order to enable the alignment of very large ontologies. Also, we have started to work on parallelization of the algorithm, by creating separate threads of execution; however, this was still not fully implemented by the time of participation in this contest. In addition, we are also working in the improvement of the general scalability of the ASMOV algorithm for the processing of ontologies with a large number of entities.

3.3 Comments on the OAEI 2009 Test Cases

The new tests added to the OAEI 2009 contests provide important and welcome tools for the improvement of ontology matching systems. Most importantly, with the availability of the instance matching tests and gold standards, and particularly the IIMB benchmarks, we have been able to redesign and thoroughly test the procedures and algorithms coded for the matching of individuals. This has resulted in a much improved version of instance matching for ASMOV. Similarly, the availability of subsumption benchmarks have also allowed us to test the corresponding algorithms. On the other hand, the continuity in the benchmark, anatomy, and conference tracks

allows us to evaluate the improvement of our algorithm and implementation as we proceed through its development.

We think it would be advantageous to count with additional gold standards for other alignments, so that the algorithms may be tested, debugged, and improved for a wider variety of conditions. Particularly, we would suggest that subsets of the mouse anatomy and NCI Thesaurus ontologies could be derived and a reference alignment provided for these subsets.

4 Conclusion

We have presented a brief description of an automated alignment tool named ASMOV, analyzed its performance at the 2009 Ontology Alignment Evaluation Initiative campaign, and compared it with its 2008 version. The test results show that ASMOV is effective in the ontology alignment realm, and because of its versatility, it performs well in multiple ontology domains such as bibliographic references (benchmark tests) and the biomedical domain (anatomy test). The tests results also showed that ASMOV is a practical tool for real-world applications that require on-the-fly alignments of ontologies.

Acknowledgments. This work is funded by the National Institutes of Health (NIH) under grant R43RR018667. The authors also wish to acknowledge the contribution of Michael Ryan, Ray Bradley, and Thomas Taylor of INFOTECH Soft, Inc.

References

1. Euzenat J and Shvaiko P. Ontology Matching. Springer-Verlag, Berlin Heidelberg, 2007.
2. Euzenat J. and Valtchev P. Similarity-based ontology alignment in OWL-lite. *In Proc. 15th ECAI*, Valencia (ES), 2004, 333-337.
3. Jean-Mary Y., Shironoshita E.P., Kabuka, M. Ontology Matching with Semantic Verification. Web Semantics: Science, Services and Agents on the World Wide Web. http://dx.doi.org/10.1016/j.websem.2009.04.001.
4. Jena from HP Labs http://www.hpl.hp.com/semweb/
5. Unified Medical Language System (UMLS) http://umlsks.nlm.nih.gov/
6. WordNet http://wordnet.princeton.edu/
7. Jean-Mary Y, Kabuka M. ASMOV: Results for OAEI 2008. http://www.dit.unitn.it/~p2p/OM-2008/oaei08_paper3.pdf. Accessed 28 Sept 2009.
8. Euzenat J, et.al. Results of the Ontology Alignment Evaluation Initiative 2007. http://www.dit.unitn.it/~p2p/OM-2007/0-o-oaei2007.pdf. Accessed 24 Sept 2008.
9. Mike Dean and Guus Schreiber, Editors, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/owl-ref/

DSSim Results for OAEI 2009

Miklos Nagy[1], Maria Vargas-Vera[1], and Piotr Stolarski[2]

[1] The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
m.nagy@open.ac.uk;m.vargas-vera@open.ac.uk
[2] Poznan University of Economics
al. Niepodleglosci 10, 60-967 Poznan, Poland
P.Stolarski@kie.ae.poznan.pl

Abstract. The growing importance of ontology mapping on the Semantic Web has highlighted the need to manage the uncertain nature of interpreting semantic meta data represented by heterogeneous ontologies. Considering this uncertainty one can potentially improve the ontology mapping precision, which can lead to better acceptance of systems that operate in this environment. Further the application of different techniques like computational linguistics or belief conflict resolution that can contribute the development of better mapping algorithms are required in order to process the incomplete and inconsistent information used and produced during any mapping algorithm. In this paper we introduce our system called "DSSim" and describe the improvements that we have made compared to OAEI 2006, OAEI 2007 and OAEI 2008.

1 Presentation of the system

1.1 State, purpose, general statement

Ontology mapping systems need to interpret heterogeneous data in order to simulate "machine intelligence", which is a driving force behind the Semantic Web. This implies that computer programs can achieve a certain degree of understanding of such data and use it to reason about a user specific task like question answering or data integration. In practice there are several roadblocks[1] that hamper the development of mapping solutions that perform equally well for different domains. Additionally the different combination of these challenges needs to be addressed in order to design systems that provides good quality results. Since DSSim has been originally designed in 2005 it has progressively evolved in order to address the combination of the 5 following challenges:

– Representation and interpretation problems: Ontology designers have a wide variety of languages and language variants to choose from in order to represent their domain knowledge. From the logical representation point of view each representations are valid separately and no logical reasoner would find inconsistency in them individually. However the problem occurs once we need to compare ontologies with different representations in order to determine the similarities between classes and individuals. Consider for example one ontology where the labels are described with standard class *rdfs:label* tag and an another ontology where the same is described

as *hasNameScientific* data property. As a result of these representation differences ontology mapping systems will always need to consider the uncertain aspects of how the semantic web data can be interpreted.

- Quality of the Semantic Web data: For every organisation or individual the context of the data, which is published can be slightly different depending on how they want to use their data. Therefore from the exchange point of view incompleteness of a particular data is quite common. The problem is that fragmented data environments like the Semantic Web inevitably lead to data and information quality problems causing the applications that process this data deal with ill-defined inaccurate or inconsistent information on the domain. The incomplete data can mean different things to data consumer and data producer in a given application scenario. Therefore applications itself need to have built in mechanisms to decide and reason about whether the data is accurate, usable and useful in essence, whether it will deliver good information and function well for the required purpose.
- Efficient mapping with large scale ontologies: Ontologies can get quite complex and very large, causing difficulties in using them for any application. This is especially true for ontology mapping where overcoming scalability issues becomes one of the decisive factors for determining the usefulness of a system. Nowadays with the rapid development of ontology applications, domain ontologies can became very large in scale. This can partly be contributed to the fact that a number of general knowledge bases or lexical databases have been and will be transformed into ontologies in order to support more applications on the Semantic Web. As a consequence applications need to scale well in case huge ontologies need to be processed.
- Task specific vs. generic systems: Existing mapping systems can clearly be classified into two categories. First group includes domain specific systems, which are build around well defined domains e.g. medical, scientific etc. These systems use specific rules, heuristics or background knowledge. As a consequence domain specific systems perform well on their own domain but their performance deteriorate across different domains. As a result the practical applicability of these systems on the Semantic Web can easily be questioned. The second group includes systems that aims to perform equally well across different domains. These systems utilise generic methods e.g. uncertain reasoning, machine learning, similarity combination etc. These systems has the potential to support a wide variety of applications on the Semantic Web in the future.

  Based on this classification it is clear that the building generic systems that perform equally well on different domains and provide acceptable results is a considerable challenge for the future research.
- Incorporating intelligence: To date the quality of the ontology mapping was considered to be an important factor for systems that need to produce mappings between different ontologies. However competitions organised on ontology mapping has demonstrated that even if systems use a wide variety techniques, it is difficult to push the mapping quality beyond certain limits. It has also been recognised [2] that in order to gain better user acceptance, systems need to introduce cognitive support for the users i.e. reduce the difficulty of understanding the presented mappings. There are different aspects of this cognitive support i.e. how to present the end

results, how to explain the reasoning behind the mapping, etc. Ongoing research focuses on how the end results can be represented in a way that end users can understand better the complex relations of large-scale ontologies. Consider for example a mapping representation between two ontologies with over 10.000 concepts each. The result file can contain thousands of mappings. To visualise this mapping existing interfaces will most likely present an unrecognizable web of connections between these properties. Even though this complex representation can be presented in a way that users could better understand the problem still arises once the users need to understand why actually these mappings have been selected. This aspect so far has totally been hidden from the end users and has formed an internal and unexpoitable part of mapping systems itself. Nevertheless in order to further improve the quality of the mapping systems these intermediary details need to be exposed to the users who can actually judge if the certain reasoning process is flawed or not. This important feedback or the ability to introspect can then be exploited by the system designers or ultimately the system itself through improving the reasoning processes, which is carried out behind the scenes in order to produce the end results. This ability to introspect the internal reasoning steps is a fundamental component of how human beings reason, learn and adapt. However, many existing ontology mapping systems that use different forms of reasoning exclude the possibility of introspection because their design does not allow a representation of their own reasoning procedures as data. Using a model of reasoning based on observable effect it is possible to test the ability of any given data structure to represent reasoning. Through such a model we present a minimal data structure[3] necessary to record a computable reasoning process and define the operations that can be performed on this representation to facilitate computer reasoning. This model facilitates the introduction and development of basic operations, which perform reasoning tasks using data recorded in this format. It is necessary that we define a formal description of the structures and operations to facilitate reasoning on the application of stored reasoning procedures. By the help of such framework provable assertions about the nature and the limits of numerical reasoning can be made.

As a result from the mapping point of view ontologies will always contain inconsistencies, missing or overlapping elements and different conceptualisation of the same terms, which introduces a considerable amount of uncertainty into the mapping process. In order to represent and reason with this uncertainty authors (Vargas-Vera and Nagy) have proposed a multi agent ontology mapping framework [4], which uses the Dempster-Shafer [5] theory in the context of Question Answering. Since our first proposition[6] of such solution in 2005 we have gradually developed and investigated multiple components of such system and participated in the OAEI in order to validate the feasibility of our proposed solution. Fortunately during the recent years our original concept has received attention from other researchers [7, 8], which helps to broaden the general knowledge on this area. We have investigated different aspects of our original idea namely the feasibility of belief combination[9] and the resolution of conflicting beliefs [10] over the belief in the correctness of similarities using the fuzzy voting model. A comprehensive description of the Fuzzy voting model can be found [10]. For this contest (OAEI 2009) the benchmarks, anatomy, directory, iimb, vlcr , Eprints-Rexa-

Sweto/DBLP benchmark and conference tracks had been tested with this new version of DSSim (v0.4).

1.2 Specific techniques used

This year within the tasks preparing the results for conference track we focused mainly on improvements and fine-tuning the algorithms for obtaining better effects in terms of both precision and recall. Moreover in order to conform to the extended terms of the track - we additionally implemented a simple enhancement for supplying subsumption correspondences as the DSSim system allowed only detection of equivalence between ontological entities. Below we will cover both types of changes more thoroughly. The first type of mentioned changes concentrates on improvements made to the compound nouns comparison method introduced in the last year's version of the system. The presented compound nouns comparison method deals with interpretation of compound nouns based on earlier works done in - among others - language understanding as well as question-answering and machine translation. The essence of the method focuses on establishing the semantic relations between items of compound nouns. During the development we reviewed some of the most interesting approaches [11] [12] [13]. Although all of them should be regarded as partial solutions, they manifest a good starting point for our experiments. Most of the cases uses either manually created rules [11] or machine learning techniques [12] in order to automatically build classification rules that will enable to rate any given compound noun phrase into one of a set of pre-selected semantic relations which best reflects the sense and nature of that phrase. We extended the initial set of simple rules by additional ones. We also made the rule engine more flexible so as it the semantic relation categories can now be assessed not only on the basis of comments or labels but also their id names. This last option is useful in some cases identified earlier in the analysis stage of the last year's results. Finally we extended also the set of semantic relation categories itself by another few categories. The compound nouns semantic relation detection algorithm is used in DSSim system as a determiner of such relations within ontology entities' identifiers, labels or comments. After the relation $r^{1,n}$ has been classified independently for entities in the first of aligned ontologies O^1 and $r^{2,m}$ separately for entities form the other ontology O^2, the alignments may be produced between the entities from O^1 and O^2 on the basis of similarity between the relations $r^{1,n}$ and $r^{2,m}$ itself. In order to eliminate the drawbacks of this approach the algorithm is viewed as a helper rather than independent factor of alignment establishment process. Nevertheless, because of the superb, multi-criterion architecture of the DSSim [14] such approach to the algorithm fits especially well allowing easy integration. As the number of elements in the set of isolated semantic relations is usually limited only to very general ones, the probability of detecting the same or similar relations is subjectively high, therefore the method itself is rather sensitive to the size of the set. Thus this year innovations concentrated on extending the rules and supplying another important categories. Moving on to another type of changes, we called the subsumption detection facility a simple one as it in fact does not alter the DSSim system algorithms to cover other types of correspondences. On the contrast the facility in this year's shape uses the results of the algorithm itself to post-produce the possible weaker (non-equivalent) correspondences basing on the algorithm result set. In order

to achieve that we implemented a straightforward inference rules over the taxonomical trees of matched ontologies. We hope to move the function to the main algorithm in the future as the simple approach introduces a number of limitations.

To sum up the introduced improvements, we made selected and subtle yet important alterations of the system. The modifications of last year proved to be useful and supplied promising results thus our intention is to build on the top of this achievements rather than starting completely different ideas. The changes introduced for this year's version of the system were backed up by the thorough interpretation and in-depth analysis of OAEI 2008 [14] outcomes.

1.3 Adaptations made for the evaluation

Our ontology mapping system is based on a multi agent architecture where each agent built up a belief for the correctness of a particular mapping hypothesis. Their beliefs are then combined into a more coherent view in order to provide better mappings. Although for the previous OAEI contests we have re-implemented our similarity algorithm as a standalone mapping process which integrates with the alignment api, we have recognised the need for possible parallel processing for tracks which contain large ontologies e.g. very large cross-lingual resources track. This need is indeed coincide with our original idea of using distributed multi-agent architecture, which is required for scalability purposes once the size of the ontology is increasing. Our modified mapping process can utilise multi core processors by splitting up the large ontologies into smaller fragments. Both the fragment size and the number of cores that should be used for processing can be set in the "param.xml" file. Based on the previous implementation we have modified our process for the OAEI 2009 which works as follows:

1. Based on the initial parameters divide the large ontologies into n*m fragments.
2. Parse the ontology fragments and submit them into the alignment job queue.
3. Run the job scheduler as long as we have jobs n the queue and assign jobs into idle processor cores.
   - 3.1 We take a concept or property from ontology 1 and consider (refer to it from now) it as the query fragment that would normally be posed by a user. Our algorithm consults WordNet in order to augment the query concepts and properties with their hypernyms.
   - 3.2 We take syntactically similar concepts and properties to the query graph from ontology 2 and build a local ontology graph that contains both concepts and properties together with the close context of the local ontology fragments.
   - 3.3 Different similarity and semantic similarity algorithms (considered as different experts in evidence theory) are used to assess quantitative similarity values (converted into belief mass function) between the nodes of the query and ontology fragment which is considered as an uncertain and subjective assessment.
   - 3.4 Then the similarity matrixes are used to determine belief mass functions which are combined using the Dempster's rule of combination. Based on the combined evidences we select those mappings in which we calculate the highest belief function.
4. The selected mappings are added into the alignment.

The overview of the mapping process is depicted on figure 1.

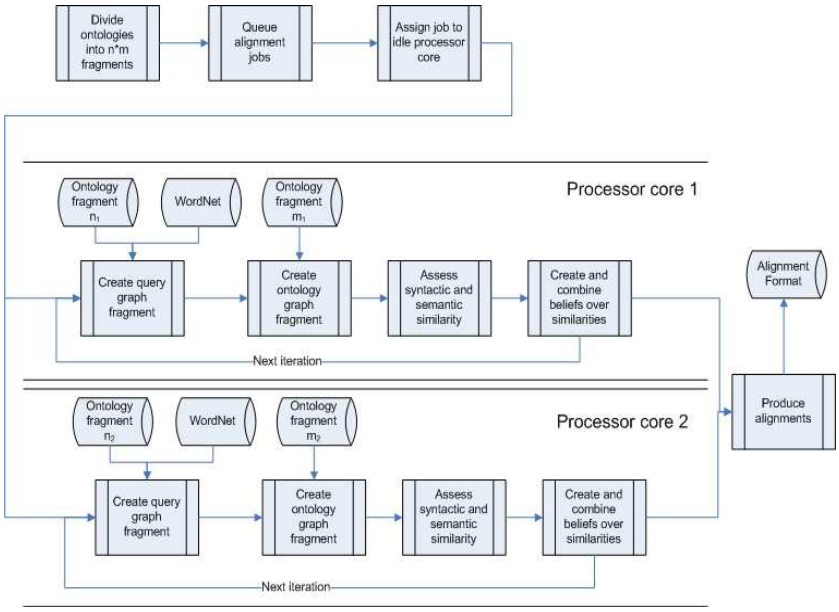


Fig. 1. The mapping process on a dual-core processor

1.4 Link to the system and parameters file

http://kmi.open.ac.uk/people/miklos/OAEI2009/tools/DSSim.zip

1.5 Link to the set of provided alignments (in align format)

http://kmi.open.ac.uk/people/miklos/OAEI2009/results/DSSim.zip

2 Results

2.1 benchmark

Our algorithm has produced the same results as last year. The weakness of our system to provide good mappings when only semantic similarity can be exploited is the direct consequence of our mapping architecture. At the moment we are using four mapping agents where 3 carries our syntactic similarity comparisons and only 1 is specialised in semantics. However it is worth to note that our approach seems to be stable compared to our last years performance, as our precision recall values were similar in spite of the fact that more and more difficult tests have been introduced in this year. As our architecture is easily expandable with adding more mapping agents it is possible to enhance our semantic mapping performance in the future. The overall conclusion is that our system produces stable quality mappings, which is good however we still see

room for improvements. Based on the 2009 results the average precision(0.97) cannot be improved significantly however considerable improvements can be made from the recall(0.66) point of view. According to the benchmarks tests our system need to be improved for cases, which contain systematic: scrambled labels + no comments + no hierarchy and systematic: scrambled labels + no comments + expanded hierarchy + no instance.

2.2 anatomy

The anatomy track contains two reasonable sized real world ontologies. Both the Adult Mouse Anatomy (2.744 classes) and the NCI Thesaurus (3.304 classes) describes anatomical concepts. The classes are represented with standard owl:Class tags with proper rdfs:label tags. Our mapping algorithm has used the labels to establish syntactic similarity and has used the rdfs:subClassOf tags to establish semantic similarities between class hierarchies. We could not make use of the owl:Restriction and oboInOwl: hasRelatedSynonym tags as this would require ontology specific additions. The anatomy track represented a number of challenges for our system. Firstly the real word medical ontologies contain classes like “outer renal medulla peritubular capillary”, which cannot be easily interpreted without domain specific background knowledge. Secondly one ontology describes humans and the second describes mice. To find semantically correct mappings between them requires deep understanding of the domain. The run time per test was around 10 min, which is an improvement compared to last year. Further we have realised significant improvement both in terms of precision and recall compared to the last year’s results. Our system ranks in the middle positions out of 10 participating systems.

2.3 Eprints-Rexa-Sweto/DBLP benchmark

This track has posed serious challenge for our system. SwetoDblp is a large-size ontology containing bibliography data of Computer Science publications where the main data source is DBLP. It contains around 1.5 million terms including 560.792 persons, 561.895 Articles in Proceedings. The eprints and rexa ontologies were large but manageable from our system’s perspective. Based on the preliminary results our system did not perform well in terms of precision and recall. The reasons needs to be investigated further. The run time including the SweetoDblp ontology was over 1 week. In spite of the fact that it was a new and difficult track this year we were disappointed with our overall results. The performance can be due to the fact that our system was originally conceived as mapping system that does not use extensively instances for establishing the mapping. As a result where only instances are present out system does not perform as well as in the other tracks.

2.4 directory

The directory test as well has been manageable in terms of execution time. In general the large number of small-scale ontologies made it possible to verify some mappings for

some cases. The tests contain only classes without any labels but in some cases different classes have been combined into one class e.g. “News_and_Media” that introduces certain level of complexity for determining synonyms using any background knowledge. To address these difficulties we have used a compound noun algorithms described in section 1.2. The execution time was around 15 minutes. In this track our performance was stable compared to the results in 2008. In terms of precision our system compares well to the other participating systems however improvements can be made from the recall point of view.

2.5 IIMB

This track contains generated benchmarks constituted using one dataset and modifying it according to various criterias. The main directory contains 37 classes and about 200 different instances. Each class contains a modified sub directory and the corresponding mapping with the instances. The different modifications introduced to the original ontology included identical copy of the original sub classes where the instance IDs are randomly changed, value transformations, structural transformations, logical transformations and several combinations of the previous transformations. The IIMB track was well manageable in terms of run time as it took under 10 minutes to run the 37 different tests. Similarly to the the task (on instance matching) described in section 2.3 our system under performed on the IIMB track. The reason for this can be attributed to the same reasons described in the E-prints-Rexa-Sweto/DBLP section.

2.6 vlcr

This vlcr track contains 3 large ontologies. The GTAA thesaurus is a Dutch public audiovisual broadcasts archive, for indexing their documents, contains around 3.800 subject keywords, 97.000 persons, 27.000 names and 14.000 locations. The DBPedia is an extremely rich dataset. It contains 2.18 million resources or ”things”, each tied to an article in the English language Wikipedia. The ”things” are described by titles and abstracts in English and often also in Dutch. We have converted the original format into standard SKOS in order to use it in our system. However we have converted only the labels in English and in Dutch whenever it was available. The third resource was the WordNet 2.0 in SKOS format where the synsets are instances rather than classes. In our system the WordNet 3.0 is included into as background knowledge therefore we have converted the original noun-synsets into a standard SKOS format and used our WordNet 3.0 as background knowledge. The run time of the track was over 1 week. Fortunately this year an other system also participated in this track therefore we can establish a qualitative comparison. In terms of precision our system performs well (name-dblp, subject-wn, location-wn, name-wn) however in certain tasks like location-dblp, person-dblp our system performs slightly worst compared to the other participating system. In terms of recall our system does not perform as well as we have expected, therefore this should be improved for the following years.

2.7 conferences

This test set is made up of collection of 15 real-case ontologies dealing with the domain of conference organization. Although all the ontologies are well embedded in the described field, nevertheless they are heterogeneous in their nature. This heterogeneity comes mainly from: designed ontology application type, ontology expressivity in terms of formalism, and robustness. Out of given 15 ontologies the production of alignments should result in 210 possible combinations (we treat the equivalent alignment as symmetric). However, we obtained 91 non-empty alignment files in the generation. From the performance point of view the alignments took about 1 hour 20 minutes on a dual core computer [3].

3 General comments

3.1 Discussions on the way to improve the proposed system

This year some tracks proved really difficult to work with. The new library track contains ontologies in different languages and due to its size first or during the mapping a translation needs to be carried out. This can be a challenge itself due to the number of concepts involved. Therefore from the background knowledge point of view we have concluded that based on the latest results that the additional multi lingual and domains specific background knowledge could provide added value for improving both recall and precision of the system.

3.2 Comments on the OAEI 2009 procedure

The OAEI procedure and the provided alignment api works very well out of the box for the benchmarks, IIMB, anatomy, directory and conference tracks. However for the Eprints-Rexa-Sweto/DBLP benchmark and vlcr and track we had to develop an SKOS parser, which can be integrated into the alignment api. Our SKOS parser convert SKOS file to OWL, which is then processed using the alignment api. Additionally we have developed a multi threaded chunk SKOS parser which can process SKOS file iteratively in chunks avoiding memory problems. For both Eprints-Rexa-Sweto/DBLP benchmark and vlcr tracks we had to develop several conversion and merging utility as the original file formats were not easily processable.

3.3 Comments on the OAEI 2009 test cases

We have found that most of the benchmark tests can be used effectively to test various aspects of an ontology mapping system since it provides both real word and generated/modified ontologies. The ontologies in the benchmark are conceived in a way that allows anyone to clearly identify system strengths and weaknesses which is an important advantage when future improvements have to be identified. The anatomy, library tests are perfect to verify the additional domain specific or multi-lingual domain knowledge. Unfortunately this year we could not integrate our system with such background knowledge so the results are not as good as we expected.

[3] Intel dual Core 3,0GHz, 512MB

4 Conclusion

Based on the experience gained during OAEI 2006, 2007, 2008 and 2009 we had a possibility to realise a measurable evolution in our ontology mapping algorithm and test it with 7 different mapping tracks. Our main objective is to improve the mapping precision with managing the inherent uncertainty of any mapping process and information in the different ontologies. The different formalisms of the ontologies suggest that on the Semantic Web there is a need to qualitatively compare and evaluate the different mapping algorithms. Participating in the Ontology Alignment Evaluation Initiative is an excellent opportunity to test and compare our system with other solutions and helped a great deal identifying the future possibilities that needs to be investigated further.

References

1. Shvaiko, P., Euzenat, J.: Ten challenges for ontology matching. Technical Report DISI-08-042, University of Trento (2008)
2. Falconer, S.M., Storey, M.A.D.: A cognitive support framework for ontology mapping. In: Proceedings of 6th International Semantic Web Conference (ISWC2007). (2007) 114–127
3. Nagy, M., Vargas-Vera, M.: Reasoning representation and visualisation framework for ontology mapping using 3d modelling. In: In Proceedings of the 4th edition of the Interdisciplinary in Engineering International Conference (Inter-Eng 2009). (2009)
4. Nagy, M., Vargas-Vera, M., Motta, E.: Dssim - managing uncertainty on the semantic web. In: Proceedings of the 2nd International Workshop on Ontology Matching. (2007)
5. Shafer, G.: A Mathematical Theory of Evidence. (1976)
6. Nagy, M., Vargas-Vera, M., Motta, E.: Multi agent ontology mapping framework in the aqua question answering system. In: International Mexican Conference on Artificial Intelligence (MICAI-2005). (2005)
7. Besana, P.: A framework for combining ontology and schema matchers with dempster-shafer (poster). In: Proceedings of the International Workshop on Ontology Matching. (2006)
8. Yaghlane, B.B., Laamari, N.: Owl-cm: Owl combining matcher based on belief functions theory. In: Proceedings of the 2nd International Workshop on Ontology Matching. (2007)
9. Nagy, M., Vargas-Vera, M., Motta, E.: Feasible uncertain reasoning for multi agent ontology mapping. In: IADIS International Conference-Informatics 2008. (2008)
10. Nagy, M., Vargas-Vera, M., Motta, E.: Managing conflicting beliefs with fuzzy trust on the semantic web. In: The 7th Mexican International Conference on Artificial Intelligence (MICAI 2008). (2008)
11. Turney, P.D.: Similarity of semantic relations. Computational Linguistics **32**(3) (2006) 379–416
12. Kim, S.N., Baldwin, T.: Interpreting semantic relations in noun compounds via verb semantics. In: Proceedings of the COLING/ACL on Main conference poster sessions. (2006) 491–498
13. Banerjee, S., Pedersen, T.: Extended gloss overlaps as a measure of semantic relatedness. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence. (2003) 805–810
14. Nagy, M., Vargas-Vera, M., Stolarski, P.: Dssim results for oaei 2008. In: Proceedings of the 3rd International Workshop on Ontology Matching. (2008)

Results of GeRoMeSuite for OAEI 2009

Christoph Quix, Sandra Geisler, David Kensche, Xiang Li

Informatik 5 (Information Systems)
RWTH Aachen University, Germany
http://www.dbis.rwth-aachen.de

Abstract. *GeRoMeSuite* is a generic model management system which provides several functions for managing complex data models, such as schema integration, definition and execution of schema mappings, model transformation, and matching. The system uses the generic metamodel *GeRoMe* for representing models, and because of this, it is able to deal with models in various modeling languages such as XML Schema, OWL, ER, and relational schemas.
A component for schema matching and ontology alignment is also part of the system. After the first participation in OAEI last year, and having established the basic infrastructure for the evaluation, we could focus this year on the improvement of the matching system. Among others, we implemented several new match strategies, such as an instance matcher and a validation method for alignments.

1 Presentation of the system

Manipulation of models and mappings is a common task in the design and development of information systems. Research in Model Management aims at supporting these tasks by providing a set of operators to manipulate models and mappings. As a framework, *GeRoMeSuite* [6] provides an environment to simplify the implementation of model management operators. *GeRoMeSuite* is based on the generic role based metamodel *GeRoMe* [5], which represents models from different modeling languages (such as XML Schema, OWL, SQL) in a generic way. Thereby, the management of models in a polymorphic fashion is enabled, i.e. the same operator implementations are used regardless of the original modeling language of the schemas. In addition to providing a framework for model management, *GeRoMeSuite* implements several fundamental operators such as Match [11], Merge [10], and Compose [8].

The matching component of *GeRoMeSuite* has been described in more detail in [11], where we present and discuss in particular the results for heterogeneous matching tasks (e.g. matching XML Schema and OWL ontologies). An overview of the complete *GeRoMeSuite* system is given in [6].

1.1 State, purpose, general statement

As a generic model management tool, *GeRoMeSuite* provides several matchers which can be used for matching models in general, i.e. our tool is not restricted to a particular domain or modeling language. Therefore, the tool provides several well known

matching strategies, such as string matchers, Similarity Flooding [9], children and parent matchers, matchers using WordNet, etc. In order to enable the flexible combination of these basic matching technologies, matching strategies combining several matchers can be configured in a graphical user interface.

Because of its generic approach, *GeRoMeSuite* is well suited for matching tasks across heterogeneous modeling languages, such as matching XML Schema with OWL. We discussed in [11] that the use of a generic metamodel, which represents the semantics of the models to be matched in detail, is more advantageous for such heterogeneous matching tasks than a simple graph representation.

Furthermore, *GeRoMeSuite* is a holistic model management and not limited to schema matching or ontology alignment. It supports also other model management tasks such as schema integration [10], model transformation [4], mapping execution and composition [7, 8].

1.2 Specific techniques used

The basis of *GeRoMeSuite* is the representation of models (including ontologies) in the generic metamodel *GeRoMe*. Any kind of model is transformed first into the generic representation, then the model management operators can be applied to the generic representation. The main advantage of this approach is that operators have to be implemented only once for the generic representation. In contrast to other (matching) approaches which use a graph representation without detailed semantics, our approach is based on the semantically rich metamodel *GeRoMe* which is able to represent modeling features in detail.

For the OAEI campaign, we focused on improving our matchers for the special case of ontology alignment, e.g. we added some features which are useful for matching ontologies. For example, the generic representation of models allows the traversal of models in several different ways. During the tests with the OAEI tasks, we realized that, in contrast to other modeling languages, traversing the ontologies using another structure than class hierarchy is not beneficial. Therefore, we configured most of our matchers that take the model structure into account just to work with the class hierarchy. Furthermore, we implemented so called 'children' and 'parent' matchers, which propagate the similarity of elements up and down in the class hierarchy.

For OAEI 2009, we added also an *Instance Matcher*, which uses instances to determine the similarity of classes and properties. Due to the flexibility and extensibility of our matching framework, the implementation of an additional matcher can be done with only a few lines of code. Basically, we just need to choose a traversal strategy which includes instances, apply one of the existing string matchers, and then choose an appropriate structural matcher to propagate the similarity of the instances to classes and properties.

In addition to last year, we also experimented with another string matcher which based on the SecondString library (`http://secondstring.sourceforge.net/`, [1]). The library provides several different string distance metrics which can be combined in various ways. The combination of 'soft' tokenization, TF-IDF based weighting of tokens, and the classical Jaro/Winkler string metric (called Soft-TFIDF) has shown good results in string matching tasks [1]. However, for the benchmarks track, we did

not find any significant difference to the string metric of [12]. For other ('real') matching tasks, the use of Soft-TFIDF might be beneficial, but we have to evaluate this with further tests.

Furthermore, we implemented a validation method using similar methods as ASMOV [3]. For difficult matching tasks with initially low values for precision and recall, the validation could increase the quality of the results by 10-20%. It is obvious, that for easy matching tasks, such as the 10x tasks in the benchmark track, the improvement cannot be so large. However, also in these tests the validation helped to achieve a perfect result.

1.3 Adaptations made for the evaluation

As only one configuration can be used for all matching tasks, we worked on strategies for measuring the quality of an alignment without having a reference alignment. We compared several statistical measures (such as expected value, variance, etc.) of alignments with different qualities in order to identify a 'good' alignment. Furthermore, these values can be used to set thresholds automatically.

During the tests, we made the experience that the expected value of all similarities, the standard deviation, and the number of mappings per model element can be used to evaluate the quality of an alignment.

Fig. 1 indicates the strategy which we used for the matching tasks in the benchmark track. All aggregation and filter steps use variable weights and thresholds, which are based on the statistical values of the input similarities.

The role matcher is a special matcher which compares the roles of model elements in our generic role-based metamodel. In principle, this results in that only elements of the same type are matched, e.g. classes with classes only and properties with properties only.

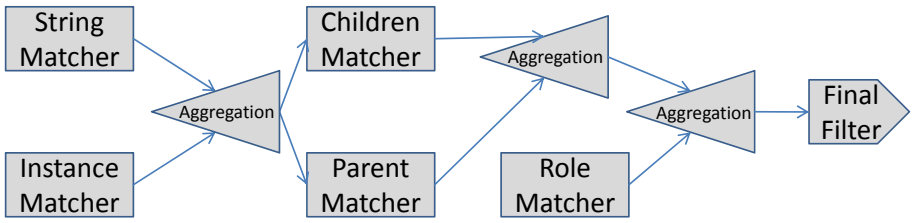


Fig. 1. Matching Strategy for OAEI

In contrast to last year, we removed Similarity Flooding [9], as it had no positive effect on the match quality. Structural similarity is already taken into account by the children and parent matchers; an additional structural matcher seems to blur the results.

On a technical level, we implemented a command line interface for the matching component, as the matching component is normally used from within the GUI frame-

work of *GeRoMeSuite*. The command line interface can work in a batch modus in which several matching tasks and configurations can be processed and compared.

1.4 Link to the system and parameters file

More information about the system can be found on the homepage of *GeRoMeSuite*: `http://www.dbis.rwth-aachen.de/gerome/oaei2009/`

The page provides also links to the configuration files used for the evaluation.

1.5 Link to the set of provided alignments (in align format)

The results for the OAEI campaign 2008 are available at `http://www.dbis.rwth-aachen.de/gerome/oaei2009/`

2 Results

2.1 Benchmark

At the cost of some performance (matching now takes about 15-25 seconds for each task instead of 5-15 as last year), our results have been significantly improved in 2009 for the benchmark track.

Overall, our matching component achieved very similar values for precision and recall, which seems to be rather unusual, if we compare our results with the results of other systems for previous years, where the precision was usually higher than recall.

Tasks 101-104 In all these very basic tasks, we achieved the perfect result.

Task	Precision	Recall 08
101	1,00	1,00
103	1,00	1,00
104	1,00	1,00

Tasks 201-210 In these tasks, the linguistic information could not always be used as labels or comments were missing. If no labels and comments are available, instance information might still help to find the right matches. We included an instance matcher in our configuration this year, which resulted in significant improvement for the 202 test cases.

Task	Precision	Recall
201	0,92	0,98
201-2	1,00	1,00
201-4	0,98	0,99
201-6	0,98	0,98
201-8	0,96	0,98
202	0,64	0,38
202-2	0,99	0,90
202-4	0,94	0,78
202-6	0,94	0,62
202-8	0,79	0,49
203	1,00	1,00
204	1,00	1,00
205	1,00	0,97
206	0,94	0,97
207	0,94	0,97
208	1,00	1,00
209	0,81	0,61
210	0,66	0,81

2.2 Tasks 221-231

The ontologies in these tasks lacked some structural information. As our matcher still uses string similarity in a first step, the results were perfect except for the case 223 for which we missed one match.

Task	Precision	Recall
221	1,00	1,00
222	1,00	1,00
223	0,99	0,99
224	1,00	1,00
225	1,00	1,00
228	1,00	1,00
230	1,00	1,00
231	1,00	1,00

Tasks 232-266 These tasks are some combinations of the tasks before. For most of the tasks, the performance of our matcher was much better than last year. However, for some matching tasks (e.g. 257, 262, 265, and 266), our system produced no result. Unfortunately, we could not resolve this problem before the deadline.

3 Comments

We participate this time the second time in OAEI and see a significant improvement of our matcher compared to last year. Thus, a structured evaluation and comparison of

ontology alignment and schema matching components as OAEI is very useful for the development of such technologies.

However, mappings between models are constructed for various reasons which can result in very different mapping results. For example, mappings for schema integration may differ from mappings for data translation. Therefore, different semantics for ontology alignments should be taken into account in the future, as it has been pointed out for schema matching in [2].

4 Conclusion

As our tool is neither specialized on ontologies nor limited to the matching task, we did not expect to deliver very good results. However, we are very satisfied with the overall results, especially compared to last year.

We will continue to work on the improvement of our matching system, especially taking into account additional validation methods, a clustering approach to handle scalability issues, and automatic methods for tuning and configuration of schema matchers. We hope to participate again with an improved system in the OAEI campaign next year.

Acknowledgements: This work is supported by the DFG Research Cluster on Ultra High-Speed Mobile Information and Communication (UMIC, `http://www.umic.rwth-aachen.de`) and by the Umbrella Cooperation Programme (`http://www.umbrella-coop.org/`).

References

1. W. W. Cohen, P. D. Ravikumar, S. E. Fienberg. A Comparison of String Distance Metrics for Name-Matching Tasks. S. Kambhampati, C. A. Knoblock (eds.), *Proc. IJCAI Workshop on Information Integration on the Web (IIWeb)*, pp. 73–78. 2003.
2. J. Evermann. Theories of Meaning in Schema Matching: A Review. *Journal of Database Management*, **19**(3):55–82, 2008.
3. Y. R. Jean-Mary, E. P. Shironoshita, M. R. Kabuka. Ontology matching with semantic verification. *Journal of Web Semantics*, **7**(3):235–251, 2009.
4. D. Kensche, C. Quix. Transformation of Models in(to) a Generic Metamodel. *Proc. BTW Workshop on Model and Metadata Management*, pp. 4–15. 2007.
5. D. Kensche, C. Quix, M. A. Chatti, M. Jarke. *GeRoMe*: A Generic Role Based Metamodel for Model Management. *Journal on Data Semantics*, **VIII**:82–117, 2007.
6. D. Kensche, C. Quix, X. Li, Y. Li. *GeRoMeSuite*: A System for Holistic Generic Model Management. C. Koch, J. Gehrke, M. N. Garofalakis, D. Srivastava, K. Aberer, A. Deshpande, D. Florescu, C. Y. Chan, V. Ganti, C.-C. Kanne, W. Klas, E. J. Neuhold (eds.), *Proceedings 33rd Intl. Conf. on Very Large Data Bases (VLDB)*, pp. 1322–1325. Vienna, Austria, 2007.
7. D. Kensche, C. Quix, X. Li, Y. Li, M. Jarke. Generic Schema Mappings for Composition and Query Answering. *Data and Knowledge Engineering*, 2009. To appear.
8. D. Kensche, C. Quix, Y. Li, M. Jarke. Generic Schema Mappings. *Proc. 26th Intl. Conf. on Conceptual Modeling (ER'07)*, pp. 132–148. 2007.
9. S. Melnik, H. Garcia-Molina, E. Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. *Proceedings of the 18th International Conference on Data Engineering (ICDE)*, pp. 117–128. IEEE Computer Society, San Jose, CA, 2002.

10. C. Quix, D. Kensche, X. Li. Generic Schema Merging. J. Krogstie, A. Opdahl, G. Sindre (eds.), *Proc. 19th Intl. Conf. on Advanced Information Systems Engineering (CAiSE'07)*, LNCS, pp. 127–141. Springer-Verlag, 2007.
11. C. Quix, D. Kensche, X. Li. Matching of Ontologies with XML Schemas using a Generic Metamodel. *Proc. Intl. Conf. Ontologies, DataBases, and Applications of Semantics (ODBASE)*, pp. 1081–1098. 2007.
12. G. Stoilos, G. B. Stamou, S. D. Kollias. A String Metric for Ontology Alignment. Y. Gil, E. Motta, V. R. Benjamins, M. A. Musen (eds.), *Proc. 4th International Semantic Web Conference (ISWC)*, *Lecture Notes in Computer Science*, vol. 3729, pp. 624–637. Springer, 2005.

KOSIMap: Ontology alignments results for OAEI 2009

Quentin Reul[1] and Jeff Z. Pan[2]

[1] VUB STARLab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
[2] University of Aberdeen, Aberdeen AB24 3FX, UK
Quentin.Reul@vub.ac.be; jeff.z.pan@abdn.ac.uk

Abstract. Ontology mapping has been recognised as an important approach to identifying similar information in heterogeneous ontologies. The Knowledge Organisation System Implicit Mapping (KOSIMap) approach relies on DL reasoning (i) to extract background knowledge about every entity, and (ii) to remove inappropriate correspondences from an alignment. The main assumption is that the use of this background knowledge reduces erroneous mappings, thus increasing coverage. In this paper, we provide an overview of KOSIMap, and present the result of our system for its first participation to the Ontology Alignment Evaluation Initiative (OAEI).

1 Presentation of KOSIMap

Ontology mapping has been recognised as an important means to identify similar information in different ontologies, thus achieving semantic interoperability on the Web. Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, the task of mapping one ontology to another is that of finding an entity in $\mathcal{O}_1$ that matches an entity in $\mathcal{O}_2$ based on their intended meaning. Many approaches to schema/ontology matching have been proposed over the years [5, 7, 10]. Furthermore, surveys reviewing these approaches, techniques and tools have been provided [4, 1]. The Knowledge Organisation System Implicit Mapping (KOSIMap) approach differs from existing approaches by relying on DL reasoning (i) to extract background knowledge about every entity, and (ii) to remove inappropriate correspondences from an alignment.

1.1 State, Purpose, General Statement

KOSIMap is an extensional and asymmetric matching approach implemented in Java. Given two consistent ontologies, KOSIMap aligns entities in the source ontology to entities in the target ontology by extracting background knowledge about entities based on DL reasoning. More specifically, a DL reasoner (e.g. FaCT++ [12], Pellet [9]) deduces logical consequences about an entity based on the asserted axioms defined in an ontology. Moreover, we investigate the use of DL reasoning to remove inappropriate correspondences from an alignment. The

main assumption is that the use of these logical consequences reduces erroneous mappings, thus increasing coverage.

The current KOSIMap implementation of produces a set of *homogeneous* correspondences, where classes are mapped to classes. object properties to object properties, and datatype properties to datatype properties. More specifically, the approach computes the similarity between two entities based on their respective sets of features (e.g. subsumption). Note that KOSIMap only considers the equivalence mapping relation between two entities.

1.2 Specific Techniques Used

The KOSIMap system calculates the similarity between entities for a pair of ontologies by analysing three features; namely lexical description (i.e. label), hierarchical structure (subsumers for concepts, and super-properties), and internal structure (inherited properties for classes, domains and ranges for object properties, and domains for datatype properties). The measures obtained by comparing these three features are then combined into a single value using a weighted sum in a similar manner to [2]. These weights are set by a user depending on the input ontologies, and requirements for the output.

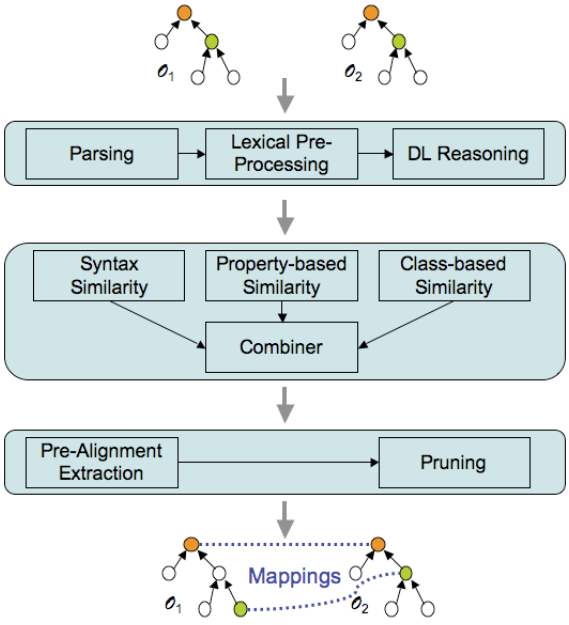


Fig. 1. The architecture of KOSIMap.

Figure 1 shows the architecture of our approach. KOSIMap consists of three main steps; namely *Pre-Processing*, *Similarity Generation*, and *Alignment Ex-*

traction. The pre-processing step includes three sub-tasks. It first parses the two ontologies with the OWL API [6]. The OWL API provides an interface to access the explicit information for each entity defined in an ontology. The API supports several representations including XML/RDF, KRSS and OBO flat files. Secondly, natural language techniques (i.e. *elimination*, *lemmatization*, and *transformation*) are applied to each entity to obtain their most basic form. Entities are not only defined by annotation properties, but also by the semantics provided by the axioms in the ontology. Thus, the final pre-processing sub-task extracts logical consequences (i.e. background information) resulting from asserted axioms. The current implementation uses the FaCT++ API[3] to classify the different ontologies.

Definition 1 (Degree of Commonality Coefficient). *Given two sets* $\mathtt{S}_s$ *and* $\mathtt{S}_t$*, the degree of commonality coefficient between them, denoted DoCCoeff(*$\mathtt{S}_s$*,* $\mathtt{S}_t$*) is defined as:*

$$DoCCoeff(S_s, S_t) = \frac{1}{\max(|S_s|, |S_t|)} \sum_{e_i \in S_s} \max_{e_j \in S_t} sim(e_i, e_j) \tag{1}$$

where S_s *is the source set,* S_t *is the target set, and* $sim(e_i, e_j)$ *computes the similarity between pair of elements in the two sets.*

Secondly, the similarity generator computes three kinds of similarities; namely *syntax similarity*, *property-based similarity*, and *class-based similarity*. The most basic feature of entities is their labels. Labels are human identifiers (i.e. words) expressed in a vocabulary shared by experts in the same domain. Therefore, we assume that equivalent classes are likely to be modelled using similar labels (or names). KOSIMap relies on string similarity (e.g. Jaro-Winkler [14], Q-Gram [11], Monge-Elkan [8], and SMOA [10]) to calculate the label similarity for each pair of entities. The SimMetrics API[4] provides a library of normalised and optimised similarity (or distance) metrics. The property-based similarity and the class-based similarity both rely on the *degree of commonality coefficient* (Definition 1) to provide an similarity value between two sets of complex objects. The property-based similarity focuses on features containing properties (i.e. set of super-properties for `OWLObjectProperty` and `OWLDataProperty` and the set of inherited properties for `OWLClass`), while the class-based similarity focuses on features containing classes (i.e. set of subsumers for `OWLClass` and the set of binary relation containing their domain and range for `OWLObjectProperty`). The results of the different similarity approaches are then aggregated for each pair of entities and stored into a n*m matrix, where is the number of element in the source ontology and m is the number of elements in the target ontology. The aggregated similarity score for a pair of entities is obtained by applying a weighted function (see Equation 2), where the weights (i.e. w_k) for each measure

[3] `http://code.google.com/p/factplusplus/`
[4] `http://sourceforge.net/projects/simmetrics/`

is in the range [0,1] and their total is 1.

$$sim(e_1, e_2) = \sum_{k=0}^{n} w_k sim_k(e_1, e_2) \quad (2)$$

After the similarity aggregation, we have a n*m matrix containing pairs of entities with a similarity value. The problem is to extract a set of relevant mappings from the matrix. This is normally achieved by discarding all candidate mappings below a threshold ζ. However, this method may return multiple mappings for each entity in the source ontology. In KOSIMap, we follow a two-step approach to extract mappings. First, the approach extracts a pre-alignment from the matrix, by selecting the maximum similarity score for each row in the matrix (i.e. for each n). This pre-alignment is then passed through a refinement process, which eliminates inappropriate mappings. In KOSIMap, we use DL reasoning to extract the local implication as part of the mapping extraction process. This approach extends the work by Wang and Xu [13], which only checked whether local implications were asserted in an ontology. As our approach only supports equivalent mapping relations, we focus on removing *inconsistent* mappings from the pre-alignment. Inconsistent mappings occur when the local consistency of an ontology is violated by the introduction of a correspondence between two ontologies. For example, an local inconsistency would occur if several entities in the source ontology are mapped to the same entity in the target ontology, and that the two classes are not recognised as equivalent by a DL reasoner.

1.3 Adaptations Made for the Evaluation

As stated in Section 1.1, KOSIMap is an asymmetric matching approach. The asymmetricity results from Equation 1, which consider the maximum value for each element in the source set. However, the organisers of OAEI campaign requested that we delivered a symmetric set of alignments. As a result, we modify the similarity generation for the property-based and class-based similarity to consider the biggest set as the source set. Moreover, we implemented a Java class to run the different tracks in batch mode. Moreover, the parameters taken by the approach (i.e. weights and thresholds) were tuned and set depending on the type of information contained in the ontologies to be mapped. For example, the property-based similarity was not calculated for the directory track as no properties were defined.

1.4 Link to the Set of Provided Alignments (in align format)

The results of the 2009 OAEI campaign for the KOSIMap system can be found at `http://www.csd.abdn.ac.uk/~qreul/research/OAEI2009.zip`.

2 Results

In this section, we present the results of the 2009 OAEI campaign obtained by the KOSIMap system. KOSIMap was used to generate alignments for four

tracks, namely benchmark, anatomy, conference and directory. Note that the full results of the Alignment Evaluation Initiative (OAEI) 2009 Campaign can be found in [3]. The experiments were carried on a Mac Book with an Intel Core 2 Duo processor (2.13GHz) and 4GB RAM running Mac OSX. The minimum memory for the Java Virtual Machine was set to 512MB, while its maximum was set to 1GB. In this experiment, we used FaCT++ as the default DL reasoner unless stated otherwise.

2.1 Benchmark

The benchmark data set contains 111 alignment tasks. KOSIMap follows the approach defined in Section 1.2. In this experiment, we used the Q-Gram similarity measure to compute the syntax similarity and as the similarity function for the degree of commonality coefficient. The FaCT++ reasoner returned an exception (`NonSimpleRoleInNumberRestrictionException`) for some tests (i.e. 222, 230, 237, 251, 258, and 304), so we used the Pellet reasoner for this test. The threshold was set to 0.2, while the weights were set as follows:

- Weight for syntax similarity: 0.3
- Weight for property-based similarity: 0.2
- Weight for class-based similarity: 0.5

KOSIMap gets near perfect alignment (Precision and Recall is 0.99) for tests 101, 103 and 104 (Table 1). Although KOSIMap performs quite well in the 2xx tests, it yields very low recall ($\leq$ 0.1) when labels in the target ontology have been scrambled (i.e. tests #202 #248, #249, #25x, and #26x). Note that KOSIMap yields high recall (i.e. $\geq$ 0.9) for tests #221 to #247. For the real ontology data set (i.e. 3xx), KOSIMap yields 0.815 for Precision and .425 for Recall. Finally, KOSIMap achieves a much better harmonic mean precision than edna even though our system yields the same recall.

Table 1. Results for KOSIMap at the OAEI 2009 campaign for the benchmark test case.

Tool	KOSIMap		edna	
Test	Prec.	Rec.	Prec.	Rec.
#1xx	0.99	0.99	0.96	1.0
#2xx	0.94	0.57	0.41	0.56
#3xx	0.72	0.50	0.47	0.82
H-Mean	0.91	0.59	0.43	0.59

2.2 Anatomy

The anatomy data set consists of two large scale anatomy ontologies. On the one hand, the Adult Mouse Anatomical Dictionary[5] represents the anatomical structure of the postnatal mouse and contains 2744 classes organised hierarchically

[5] `http://www.informatics.jax.org/searches/AMA_form.shtml`

by "is-a" and "part-of" relationships. On the other hand, the NCI Thesaurus[6] is a reference terminology and biomedical ontology covering clinical care, translational and basic research, public information, and administrative activities. This ontology contains a subset of the classes defined in the thesaurus (i.e. 3304 classes). Note that the property-based similarity was discarded for this track as these ontologies only contain a very small number of properties.

KOSIMap produces an alignment for three of the four sub-tasks of this track:

1. *Optimal solution*: The optimal solution is obtained with a threshold set to 0.6, the syntax similarity set to 0.6 and the class-based similarity set to 0.4. It took KOSIMap approximately 5 min to generate the alignment.
2. *Optimal precision*: The optimal solution is obtained with a threshold set to 0.7, the syntax similarity set to 0.6 and the class-based similarity set to 0.4. It took KOSIMap approximately 5 min to generate the alignment.
3. *Optimal recall*: The optimal solution is obtained with a threshold set to 0.6, the syntax similarity set to 0.6 and the class-based similarity set to 0.4. It took KOSIMap approximately 5 min to generate the alignment.

Table 2. Results for KOSIMap at the OAEI 2009 campaign for the anatomy test case.

Tool	Optimal solution				Optimal precision			Optimal recall		
	Runtime	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.
KOSIMap	≈ 5 min	0.87	0.62	0.72	0.91	0.45	0.60	0.87	0.62	0.72

The results of the anatomy track are shown in Table 2. KOSIMap takes around 5 minutes to extract mappings between the Adult Mouse Anatomical Dictionary and the NCI Thesaurus and the F-Measure is 0.72. We observer significant differences with regard to the trade-off between precision and recall. For instance, we observe that the recall obtained by KOSIMap falls from 0.62 to 0.45 when generating an alignment for optimal precision sub-task. As KOSIMap favours recall over precision, the score obtained for the optimal recall sub-task is the same as the optimal solution.

2.3 Conference

This track contains 15 ontologies covering the conference organization domain. These ontologies differ in terms of DL expressivity and size. For example, *ekaw.owl* is represented in $\mathcal{SHIN}$, while *paperdyne.owl* is expressed in $\mathcal{ALCHIN}$(D).

KOSIMap generated 105 non-empty alignments with parameters set as follows:

- Weight for syntax similarity: 0.3
- Weight for property-based similarity: 0.2
- Weight for class-based similarity: 0.5

[6] `http://www.cancer.gov/cancerinfo/terminologyresources/`

Table 3 shows the precision, recall, and F-measure computed for three different thresholds (0.2, 0.5, and 0.7). The results show that KOSIMap reaches an optimal solution with the threshold set to 0.5 before obtaining lower performances with higher thresholds. Moreover, the precision achieved by our system increases at the same time as the threshold.

Table 3. Results for KOSIMap at the OAEI 2009 campaign for the conference test case.

Tool	threshold=0.2			threshold=0.5			threshold=0.7		
	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.	Prec.	Rec.	FMeas.
KOSIMap	0.18	0.56	0.27	0.41	0.43	0.41	0.70	0.23	0.33

2.4 Directory

The directory track consists of 4639 test cases. As no properties (object properties or datatype properties) are found in this track, the property-based similarity is discarded for this track. In this experiment, the threshold is set to .0, while the weights are set to 0.6 (for syntax similarity) and 0.4 (for the class-based similarity). Due of the low expressivity of the ontologies (i.e. $\mathcal{AL}$), we simplified the rules to retain the correspondence with the highest score when a class in the source ontology maps to several classes in the target ontology. KOSIMap takes just over 1 minute to generate the 4639 alignments. The preliminary results of this track yielded a score of 0.618 for Precision, 0.453 for Recall, and a F-Measure of 0.523.

3 General Comments

3.1 Comments on the Results

From the results we can see that KOSIMap can take advantage of all different features associated with entities. The lexical description is especially important to achieve high precision and recall, while the hierarchical and internal structure are used to refine the final alignment. For example, tests in the benchmark track with scrambled labels (i.e. tests 248 to 266) tend to yield very low recall.

Based on the anatomy track, we have demonstrated the scalability of our approach. Although the two ontologies are not very expressive (i.e. $\mathcal{ALE}$ for AMA and $\mathcal{ALE}+$ for the NCI thesaurus), we have shown that the use of a DL reasoner does not impact the scalability of our system. Thus, this result suggests that the use of a reasoner does not greatly increase the runtime of the mapping task. Note that testing on more expressive large-scale ontologies should be carried to further test this observation.

3.2 Discussions on the Way to Improve KOSIMap

KOSIMap uses different strategies to extract correspondences between two ontologies. Based on the test library, we have seen that some strategies (e.g. property-based similarity) were not always useful to extract alignments. One possible way to improve the current system would be to include a strategy selection module. With strategy selection, KOSIMap could avoid some some noise produced by some strategies when the information these strategies rely on is not adequate. For example, when no properties are defined in the ontology.

Another improvement to the system would be to include a module to fine-tune weights when combining the different similarity measure. The current approach relies on the user to analyse the information contained in the ontologies. It is important to note that this is process is both time-consuming and error prone. A solution to this problem would be to consider the DL expressivity of both ontologies to analyse the impact of each measure on the global similarity value.

3.3 Comments on the OAEI 2009 Test Cases

The advantage of the OAEI test library is that it provides a wide range of tests covering real word and modied ontologies. For example, the *benchmark* track allows anyone to clearly identify the strengths and weaknesses of their systems. The library also includes test cases for comparing large scale ontologies. However, the ontologies provided in the anatomy track are not very expressive. As a result, it is difficult to address the impact of using DL reasoners on large scale ontologies.

4 Conclusion

In this paper, we present the KOSIMap system, which aligns entities from two ontologies. This system relies on DL reasoning to (i) extract background knowledge about every entity, and (ii) to remove inappropriate correspondences from an alignment. KOSIMap consists of three main steps; namely *Pre-Processing*, *Similarity Generation*, and *Alignment Extraction*. It first parses the two ontologies, extracts the implicit structure of both ontologies using an OWL DL reasoner, and applies natural language techniques to lexical descriptions (i.e. labels). Next, it computes three different types of similarities for every pair of entities. These similarity values are then combined and stored in a n*m matrix from which a pre-alignment is extracted. This pre-alignment is then passed through a refinement process, which eliminates inconsistent mappings.

Secondly, we report the results obtained by KOSIMap for its first participation to the Ontology Alignment Evaluation Initiative. From the results of the benchmark test case, we can see that our system can take advantage of all different features associated with entities during the ontology mapping task. We have also shown that KOSIMap remains scalable despite using DL reasoning throughout the mapping process. However, testing on more expressive large-scale ontologies should be carried to further test this observation.

5 Acknowledgements

The IPAS project is co-funded by the Technology Strategy Board's Collaborative Research and Development programme (www.innovateuk.org) and Rolls-Royce (project No. TP/2/IC/6/I/10292).

References

1. S. Castano, A. Ferrara, S. Montanelli, G. N. Hess, and S. Bruno. State of the art on ontology coordination and matching. Report FP6-027538, BOEMIE, March 2007.
2. M. Ehrig and Y. Sure. Ontology mapping - an integrated approach. In *Proceedings of the 1st European Semantic Web Symposium (ESWS 04)*, pages 76–91, 2004.
3. J. Euzenat, A. Ferrara, L. Hollink, V. Malaise, C. Meilicke, A. Nikolov, J. Pane, F. Scharffe, P. Shvaiko, V. Spiliopoulos, H. Stuckenschmidt, O. Svab-Zamazal, V. Svatek, C. T. dos Santos, and G. Vouros. Preliminary results of the ontology alignment evaluation initiative 2009. In *Proceedings of the 4th International Workshop on Ontology Matching (OM-2009)*, 2009.
4. J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer-Verlag, Berlin, 2007.
5. J. Euzenat and P. Valtchev. An integrative proximity measure for ontology alignment. In *Proceedings of ISWC-2003 Workshop on Semantic Information Integration*, page 3338, 2003.
6. M. Horridge, S. Bechhofer, and O. Noppens. Igniting the OWL 1.1 touch paper: The OWL API. In *Proceedings of the 3rd OWL Experienced and Directions Workshop (OWLED 2007)*, Innsbruck, Austria, 2007.
7. A. Maedche and S. Staab. Measuring similarity between ontologies. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002)*, pages 251–263, Siguenza, Spain, 2002.
8. A. E. Monge and C. Elkan. The field matching problem: Algorithms and applications. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.
9. E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53, June 2007.
10. G. Stoilos, G. Stamou, and S. Kollias. A string metric for ontology alignment. In *Proceeding of the 4th International Semantic Web Conference (ISWC 2005)*, 2005.
11. E. Sutinen and J. Tarhio. On using Q-Gram locations in approximate string matching. In *Proceedings of the 3rd Annual European Symposium on Algorithms (ESA 95)*, page 327340, 1995.
12. D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR 2006)*, pages 292–297, Seattle, USA, August 2006.
13. P. Wang and B. Xu. Debugging ontology mappings: A static approach. *Computing and Informatics*, 22:1001–1015, 2003.
14. W. E. Winkler. String comparator metrics and enhanced decision rules in the Fellegi-Sunter Model of record linkage. In *Proceedings of the Section on Survey Research Methods*, pages 354–359. American Statistical Association, 1990.

Lily: Ontology Alignment Results for OAEI 2009

Peng Wang[1], Baowen Xu[2,3]

[1]College of Software Engineering, Southeast University, China
[2]State Key Laboratory for Novel Software Technology, Nanjing University, China
[3]Department of Computer Science and Technology, Nanjing University, China
pwang@seu.edu.cn, bwxu@nju.edu.cn

Abstract. This paper presents the alignment results of Lily for the ontology alignment contest OAEI 2009. Lily is an ontology mapping system, and it has four functions: generic ontology matching, large scale ontology matching, semantic ontology matching and mapping debugging. In OAEI 2009, Lily submited the results for four alignment tasks: benchmark, anatomy, directory and conference.

1 Presentation of the system

Lily is an ontology mapping system for solving the key issues related to heterogeneous ontologies, and it uses hybrid matching strategies to execute the ontology matching task. Lily can be used to discovery the mapping for both normal ontologies and large scale ontologies. In the past year, we did not improve Lily significantly but revised some bugs according to the reports from some users.

1.1 State, purpose, general statement

In order to obtain good alignments, the core principle of the matching strategy in Lily is utilizing the useful information effectively and rightly. Lily combines several novel and efficient matching techniques to find alignments. Currently, Lily realized four main functions: (1) Generic Ontology Matching method (GOM) is used for common matching tasks with small size ontologies. (2) Large scale Ontology Matching method (LOM) is used for the matching tasks with large size ontologies. (3) Semantic Ontology Matching method (SOM) is used for discovering the semantic relations between ontologies. Lily uses the web knowledge to recognize the semantic relations through the search engine. (4) Ontology mapping debugging is used to improve the alignment results.

The matching process mainly contains three steps: (1) In preprocess, Lily parses ontologies and prepares the necessary data for the subsequent steps. (2) In computing step, Lily uses suitable methods to calculate the similarity between elements from different ontologies. (3)In post-process, the alignments are extracted and then refined by mapping debugging. The architecture of Lily is shown in Fig. 1.

The lasted version of Lily is V2.0. Lily V2.0 provides a friendly graphical user interface. Fig.2 shows a snapshot when Lily is running.

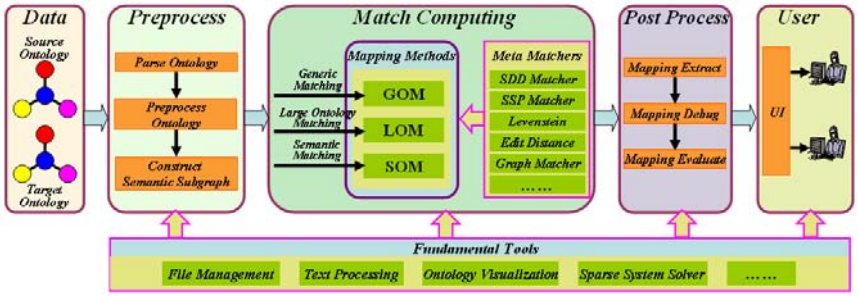


Fig. 1. The Architecture of Lily

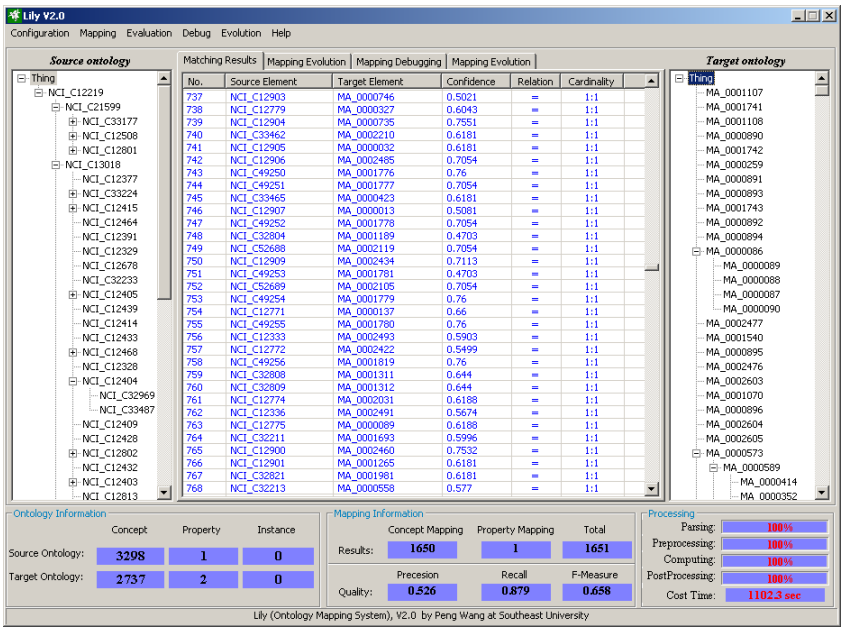


Fig. 2. The user interface of Lily

1.2 Specific techniques used

Lily aims to provide high quality 1:1 alignments between concept/property pairs. The main specific techniques used by Lily are as follows.

Semantic subgraph An entity in a given ontology has its specific meaning. In our ontology mapping view, capturing such meaning is very important to obtain good alignment results. Therefore, before similarity computation, Lily first describes the meaning for each entity accurately. The solution is inspired by the method proposed by Faloutsos et al. for discovering connection subgraphs [1]. It is based on electricity

analogues to extract a small subgraph that best captures the connections between two nodes of the graph. Ramakrishnan et al. also exploits such idea to find the informative connection subgraphs in RDF graph [2].

The problem of extracting semantic subgraphs has a few differences from Faloutsos's connection subgraphs. We modified and improved the methods provided by the above two work, and proposed a method for building an *n-size* semantic subgraph for a concept or a property in ontology. The subgraphs can give the precise descriptions of the meanings of the entities, and we call such subgraphs semantic subgraphs. The detail of the semantic subgraph extraction process is reported in our other work [3].

The significance of semantic subgraphs is that we can build more credible matching clues based on them. Therefore it can reduce the negative affection of the matching uncertain.

Generic ontology matching method The similarity computation is based on the semantic subgraphs, i.e. all the information used in the similarity computation is come from the semantic subgraphs. Lily combines the text matching and structure matching techniques [3].

Semantic Description Document (SDD) matcher measures the literal similarity between ontologies. A semantic description document of a concept contains the information about class hierarchies, related properties and instances. A semantic description document of a property contains the information about hierarchies, domains, ranges, restrictions and related instances. For the descriptions from different entities, we calculate the similarities of the corresponding parts. Finally, all separate similarities are combined with the experiential weights. For the regular ontologies, the SDD matcher can find satisfactory alignments in most cases.

To solve the matching problem without rich literal information, a similarity propagation matcher with strong propagation condition (SSP matcher) is presented, and the matching algorithm utilizes the results of literal matching to produce more alignments. Compared with other similarity propagation methods such as similarity flood [4] and SimRank [5], the advantages of our similarity propagation include defining stronger propagation condition, semantic subgraphs-based and with efficient and feasible propagation strategies. Using similarity propagation, Lily can find more alignments that cannot be found in the text matching process.

However, the similarity propagation is not always perfect. When more alignments are discovered, more incorrect alignments would also be introduced by the similarity propagation. So Lily also uses a strategy to determine when to use the similarity propagation.

Large scale ontology matching Large scale ontology matching tasks propose the rough time complexity and space complexity for ontology mapping systems. To solve this problem, we proposed a novel method [3], which uses the negative anchors and positive anchors to predict the pairs can be passed in the later matching computing. The method is different from other several large scale ontology matching methods, which are all based on ontology segment or modularization.

Semantic ontology matching Our semantic matching method [6] is base on the idea that Web is a large knowledge base, and from which we can gain the semantic relations between ontologies through Web search engine. Based on lexico-syntactic patterns, this method first obtains a candidate mapping set using search engine. Then

the candidate set is refined and corrected with some rules. Finally, ontology mappings are chosen from the candidate mapping set automatically.

Ontology mapping debugging Lily uses a technique called ontology mapping debugging to improve the alignment results [7]. During debugging, some types of mapping errors, such as redundant and inconsistent mappings, can be detected. Some warnings, including imprecise mappings or abnormal mappings, are also locked by analyzing the features of mapping result. More importantly, some errors and warnings can be repaired automatically or can be presented to users with revising suggestions.

1.3 Adaptations made for the evaluation

In OAEI 2009, Lily used GOM matcher to compute the alignments for three tracks (benchmark, directory, conference). In order to assure the matching process is fully automated, all parameters are configured automatically with a strategy. For the large ontology alignment tracks (anatomy), Lily used LOM matcher to discover the alignments. Lily can determine which matcher should be chose according to the size of ontology.

1.4 Link to the system and the set of provided alignments

Lily V2.0 and the alignment results for OAEI 2009 are available at http://ontomappinglab.googlepages.com/lily.htm.

2 Results

2.1 benchmark

The benchmark test set can be divided into five groups: 101-104, 201-210, 221-247, 248-266 and 301-304.

The following table shows the average performance of each group and the overall performance on the benchmark test set.

Table 1. The performance on the benchmark

	101-104	201-210	221-247	248-266	301-304	Average	H-mean
Precision	1.00	0.99	0.99	0.94	0.83	0.95	0.97
Recall	1.00	0.95	1.00	0.76	0.79	0.84	0.88

2.2 anatomy

The anatomy track consists of two real large-scale biological ontologies. Lily can handle such ontologies smoothly with LOM method. Lily submitted the results for three sub-tasks in anatomy. Task#1 means that the matching system has to be applied with standard settings to obtain a result that is as good as possible. Task#2 means that

the system generates the results with high precision. Task#3 means that the system generates the alignment with high recall.

2.3 directory

The directory track requires matching two taxonomies describing the web directories. Except the class hierarchy, there is no other information in the ontologies. Therefore, besides the literal information, Lily also utilizes the hierarchy information to decide the alignments.

2.4 conference

This task contains 15 real-case ontologies about conference. For a given ontology, we compute the alignments with itself, as well as with other ontologies. For we treat the equivalent alignment is symmetric, we get 105 alignment files totally. The heterogeneous character in this track is various. It is a challenge to generate good results for all ontology pairs in this test set.

3 General comments

Strengths For normal size ontologies, if they have regular literals or similar structures, Lily can achieve satisfactory alignments.

Weaknesses Lily needs to extract semantic subgraphs for all concepts and properties. It is a time-consuming process. Even though we have improved the efficiency of the extracting algorithm, it still is the bottleneck for the performance of the system.

4 Conclusion

We briefly introduce our ontology matching tool Lily. The matching process and the special techniques used by Lily are presented. The preliminary alignment results are carefully analyzed. Finally, we summarized the strengths and the weaknesses of Lily.

References

1. Faloutsos, C., McCurley, K. S., Tomkins, A.: Fast Discovery of Connection Subgraphs. In the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington (2004).
2. Ramakrishnan, C., Milnor, W. H., Perry, M., Sheth, A. P.: Discovering Informative Connection Subgraphs in Multirelational Graphs. ACM SIGKDD Explorations, Vol. 7(2), (2005)56-63.
3. Wang, P.: Research on the Key Issues in Ontology Mapping (In Chinese). PhD Thesis, Southeast University, Nanjing, 2009.
4. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. In the 18th International Conference on Data Engineering (ICDE), San Jose CA (2002).

5. Jeh, G., Widom, J.: SimRank: A Measure of Structural-Context Similarity. In the 8th International Conference on Knowledge Discovery and Data Mining (SIGKDD), Edmonton, Canada, (2002).
6. Li, K., Xu, B., and Wang, P. An Ontology Mapping Approach Using Web Search Engine. Journal of Southeast University, 2007, 23(3):352-356.
7. Wang, P., Xu, B. Debugging Ontology Mapping: A Static Method. Computing and Informatics, 2008, 27(1): 21–36.

Appendix: Raw results

The final results of benchmark task are as follows.

Matrix of results

#	Comment	Prec.	Rec.	#	Comment	Prec.	Rec.
101	Reference alignment	1.00	1.00	251		0.96	0.76
103	Language generalization	1.00	1.00	251-2		0.99	0.96
104	Language restriction	1.00	1.00	251-4		0.99	0.90
201	No names	1.00	1.00	251-6		0.96	0.84
201-2		1.00	1.00	251-8		0.99	0.83
201-4		1.00	1.00	252		0.95	0.77
201-6		1.00	1.00	252-2		0.98	0.94
201-8		1.00	1.00	252-4		0.98	0.94
202	No names, no comment	1.00	0.84	252-6		0.98	0.94
202-2		1.00	0.95	252-8		0.97	0.93
202-4		1.00	0.92	253		0.85	0.62
202-6		0.98	0.88	253-2		1.00	0.93
202-8		0.98	0.84	253-4		1.00	0.91
203	Misspelling	1.00	0.98	253-6		0.94	0.82
204	Naming conventions	1.00	1.00	253-8		0.98	0.82
205	Synonyms	1.00	0.99	254		1.00	0.27
206	Translation	1.00	0.99	254-2		1.00	0.82
207		1.00	0.99	254-4		1.00	0.70
208		1.00	0.98	254-6		1.00	0.61
209		0.97	0.87	254-8		1.00	0.42
210		1.00	0.88	257		1.00	0.12
221	No hierarchy	1.00	1.00	257-2		1.00	0.97
222	Flattened hierarchy	1.00	1.00	257-4		1.00	0.94
223	Expanded hierarchy	0.98	0.97	257-6		0.87	0.82
224	No instances	1.00	1.00	257-8		0.85	0.67
225	No restrictions	1.00	1.00	258		0.76	0.56
228	No properties	1.00	1.00	258-2		0.99	0.96
230	Flattening entities	0.94	1.00	258-4		0.96	0.88
231	Multiplying entities	1.00	1.00	258-6		0.95	0.83

232	No hierarchy no instance	1.00	1.00	258-8		0.94	0.80
233	No hierarchy no property	1.00	1.00	259		0.91	0.73
236	No instance no property	1.00	1.00	259-2		0.97	0.94
237		1.00	1.00	259-4		0.97	0.94
238		0.98	0.98	259-6		0.96	0.93
239		0.97	1.00	259-8		0.97	0.94
240		0.97	1.00	260		0.94	0.55
241		1.00	1.00	260-2		0.93	0.93
246		0.97	1.00	260-4		0.90	0.93
247		0.94	0.97	260-6		0.93	0.86
248		1.00	0.81	260-8		0.95	0.69
248-2		1.00	0.95	261		0.61	0.33
248-4		1.00	0.92	261-2		0.88	0.91
248-6		1.00	0.88	261-4		0.88	0.91
248-8		1.00	0.87	261-6		0.88	0.91
249		0.76	0.73	261-8		0.88	0.91
249-2		1.00	0.97	262		NaN	0.00
249-4		0.98	0.91	262-2		1.00	0.76
249-6		0.98	0.87	262-4		1.00	0.61
249-8		0.95	0.82	262-6		1.00	0.42
250		1.00	0.55	262-8		1.00	0.21
250-2		1.00	1.00	265		0.80	0.14
250-4		1.00	1.00	266		0.50	0.09
250-6		1.00	1.00	301	BibTeX/MIT	0.87	0.81
250-8		0.90	0.79	302	BibTeX/UMBC	0.84	0.65
				303	Karlsruhe	0.63	0.75
				304	INRIA	0.96	0.96

MapPSO Results for OAEI 2009

Jürgen Bock[1], Peng Liu[1], and Jan Hettenhausen[2]

[1] FZI Forschungszentrum Informatik an der Universität Karlsruhe, Germany
{bock, pengliu}@fzi.de
[2] Griffith University, Institute for Integrated and Intelligent Systems, Brisbane, Australia
j.hettenhausen@griffith.edu.au

Abstract. This paper presents and discusses the results of the latest developments of the MapPSO system, which is an ontology alignment approach that is based on discrete particle swarm optimisation. Firstly it is recalled, how the algorithm approaches the ontology matching task as an optimisation problem, and how the specific technique of particle swarm optimisation is applied. Secondly, the results are discussed, which were achieved for the Benchmark data set of the 2009 Ontology Alignment Evaluation Initiative.

1 Presentation of the system

With last year's OAEI campaign the MapPSO system (Ontology **Map**ping by **P**article **S**warm **O**ptimisation) has been introduced [1] as a novel research prototype, which is expected to become a highly scalable, massively parallel tool for ontology alignment. In the following subsection the basic idea of this approach will be sketched.

1.1 State, purpose, general statement

The MapPSO algorithm is being developed for the purpose of aligning large ontologies. It is motivated by the observation that ontologies and schema information such as thesauri or dictionaries are not only getting numerous on the web, but also are becoming increasingly large in terms of the number of classes/concepts and properties/relations. This development raises the need for highly scalable tools to provide interoperability and integration of various heterogeneous sources. On the other hand the emergence of parallel architectures provide the basis for highly parallel and thus scalable algorithms which need to be adapted to these architectures.

The presented MapPSO method regards the ontology alignment problem as an optimisation problem which allows for the adaptation of a discrete variant of particle swarm optimisation [2, 3], a population based optimisation paradigm inspired by social interaction between swarming animals. Particularly the population based structure of this method provides high scalability on parallel systems. Particle swarm optimisation furthermore belongs to the group of anytime algorithms, which allow for interruption at any time and will provide the best answer being available at that time. Particularly this property might be interesting when an alignment problem is subject to certain time constraints.

Compared to the first version of the system that participated in last year's OAEI campaign, some adaptation have been made with particular respect to the base matchers used. More precisely, the existing base matchers have been improved, and new base matchers have been applied, in order to improve the quality of the alignments discovered by MapPSO. Section 2 shows the improvements compared to OAEI 2008.

1.2 Specific techniques used

MapPSO utilises a discrete particle swarm optimisation (DPSO) algorithm, based in parts on the DPSO developed by Correa *et al.* [2, 3], to tackle the ontology matching problem as an optimisation problem. The core element of this optimisation problem is the objective function which supplies a fitness value for each candidate alignment. To find solutions for the optimisation problem, MapPSO simulates a set of particles whereby each particle is a candidate alignment comprising a set of initially random mappings. (Currently only 1:1 alignments are supported.) Each of these particles maintains a memory of previously found good mappings (*personal best*) and the swarm maintains a collective memory of the best known alignment so far (*global best*). In each iteration, particles are updated by changing their sets of correspondences in a guided random manner. Correspondences which are also present in the global best set and personal best set are more likely to be kept, as are those with a very good evaluation. Worst Correspondences are more likely to be removed and replaced with other correspondences which are random recommended from best alignment(*personal best* and *global best*) and random created according to left available entities. Each candidate alignment of two ontologies is scored based on the sum of quality measures of the single correspondences. The currently best alignment is the one with the best known fitness rating according to these criteria. According to this revisit of the ontology matching problem, a particle swarm can be applied to search for the optimal alignment.

For each correspondence the quality score is calculated based on an aggregation of scores from a configurable set of base matchers. Each base matcher provides a distance measure for each correspondence. Currently the following base matchers are used:

- SMOA string distance [4] for entity names
- SMOA string distance for entity labels
- WordNet distance for entity names
- WordNet distance for entity labels
- Vector space similarity [5] for entity comments
- Hierarchy distance to propagate similarity of super/subclasses and super/subproperties
- Structural similarity of classes derived from properties that have them as domain or range classes
- Structural similarity of properties derived from their domain and range classes
- Similarity of classes derived from individuals that are instances of them
- Similarity of properties derived from individuals that are subjects or objects of them
- Similarity of individuals derived from property assertions, in particular the following:
  - values of data properties, the resp. individual is asserted to

- object (individuals) of object properties, the resp. individual is asserted to as subject
- subject (individuals) of object properties, the resp. individual is asserted to as object

For each correspondence the available base distances are aggregated by applying a weighted average operator. Hereby a fixed weight is assigned to each base distance. However, the weight configuration is automatically adjusted before the alignment process, according to the ontology characteristics. By this analysis those characteristics are determined that are most promising for detecting similarities. The evaluation of the overall alignment of each particle is computed by aggregating all its correspondence distances. In the current implementation each particle runs in a separate thread and all fitness calculations and particle updates are performed in parallel. The only sequential portion on the algorithm is the synchronisation after each iteration to acquire the fitness value from each particle and determine the currently global best alignment.

1.3 Adaptations made for the evaluation

Since MapPSO is an early prototype, the OAEI Benchmark test data is used during the development process. No specific adaptations have been made.

1.4 Link to the system and parameters file

The release of MapPSO (`MapPSO.jar`) and the parameter file `params.xml` used for OAEI 2009 are located at `https://sourceforge.net/projects/mappso/files/` in the folder `oaei2009`.

1.5 Link to the set of provided alignments (in align format)

The alignments of the OAEI 2009 benchmark data set as provided by MapPSO are located in the file `alignments.zip` at `https://sourceforge.net/projects/mappso/files/`.

2 Results

The MapPSO system participated only in the benchmarks track this year.

The algorithm is highly adjustable via its parameter file and can be tuned to perform well on specific problems, as well as to perform well for precision or recall. To obtain the results presented in Tab. 1 a compromised parameter configuration was used.

2.1 benchmark

The Benchmark test case is designed to provide a number of data sets systematically revealing strengths and weaknesses of the matching algorithm. In the case of MapPSO the experiences were as follows.

Note, that in the results where computed without consulting WordNet in order to improve run-time performance.

For tests **101–104** MapPSO achieves precision and recall values of 100 %. Since the ontologies in those tests have complete information, which can used for alignment. The results have slightly improved compared to the results from 2008.

As for tests **201–210** results are slightly worse than for tests 101–104, since by each test, one or more types of linguistic information are lost, so the system has to rely on other information and on different base matchers resp. in order to determine the similarity of entities. The quality of the alignment decreases with the number of features that provide linguistic features to exploit. In particular for test 202, all names, labels and comments are unavailable, the system achieves about 63 % precision and recall by using solely structural and semantic information. However, with newly added base matchers which respect ABox information in ontologies, the results for tests 201-210 are much improved as last year.

In tests **221–247**, where the structure of the ontologies varies, the results are similar to the 10x tests. Since the linguistic features can be used by MapPSO, which is still the main focus of the current implementation of MapPSO.

The tests **248–266** combine linguistic and structural problems. As the results show, the quality of the alignments is decreasing with the decreasing number of features available in the ontologies. The results of some tests are slightly worse as 2008, for instance 249-2. The reason is possibly the using of weighted average operator instead of ordered weighted average operator and deactivating WordNet distance.

For the real-world tests **301–304**, no uniform results can be derived as the algorithm's precision and recall values vary between 0 and 60 %.

All together, results of our system MapPSO in 2009 is significantly improved compared to the previous version in 2008, but since the test is run without WordNet there are some tests with worse results.

3 General comments

In the following we will provide a few statements on our experiences from participating in the OAEI 2008 competition and briefly discuss future work on the MapPSO algorithm.

3.1 Comments on the results

Firstly it shall be noted that MapPSO is a non-deterministic method and therefore on a set of independent runs the quality of the results and the number of mappings in the alignments will be subject to slight fluctuations.

3.2 Discussions on the way to improve the proposed system

With the latest version of MapPSO several new base matchers have been applied in the system, which significantly improved the quality of the results. In particular, the system makes use of *lexical*, *linguistic*, *structural*, and to a certain extent *semantic*

Table 1. Results of MapPSO in the OAEI 2009 benchmark data set.

Test Name	Precision	Recall	Test Name	Precision	Recall	Test Name	Precision	Recall
101	1	1	246	0.97	1	257	0.24	0.24
103	1	1	247	0.85	0.88	257-2	0.88	0.88
104	1	1	248	0.61	0.61	257-4	0.94	0.94
201	1	1	248-2	0.61	0.61	257-6	0.61	0.61
201-2	1	1	248-4	0.58	0.58	257-8	0.52	0.52
201-4	0.98	0.98	248-6	0.58	0.58	258	0.1	0.1
201-6	1	1	248-8	0.58	0.58	258-2	0.28	0.28
201-8	1	1	249	0.04	0.04	258-4	0.17	0.17
202	0.64	0.64	249-2	0.3	0.3	258-6	0.07	0.08
202-2	0.94	0.94	249-4	0.23	0.23	258-8	0.12	0.12
202-4	0.7	0.7	249-6	0.12	0.12	259	0.04	0.04
202-6	0.86	0.86	249-8	0.1	0.1	259-2	0.23	0.23
202-8	0.69	0.69	250	0.39	0.39	259-4	0.22	0.22
203	1	1	250-2	1	1	259-6	0.23	0.23
204	1	1	250-4	0.79	0.79	259-8	0.21	0.21
205	1	0.99	250-6	0.55	0.55	260	0.13	0.14
206	1	0.99	250-8	0.48	0.48	260-2	0.77	0.79
207	1	0.99	251	0.58	0.58	260-4	0.6	0.62
208	0.97	0.97	251-2	0.87	0.87	260-6	0.37	0.38
209	0.68	0.67	251-4	0.72	0.72	260-8	0.33	0.34
210	0.7	0.7	251-6	0.58	0.58	261	0.12	0.12
221	1	1	251-8	0.57	0.57	261-2	0.47	0.48
222	1	1	252	0.45	0.45	261-4	0.59	0.61
223	0.97	0.97	252-2	0.77	0.77	261-6	0.53	0.55
224	1	1	252-4	0.76	0.76	261-8	0.5	0.52
225	1	1	252-6	0.77	0.77	262	0.06	0.06
228	1	1	252-8	0.84	0.84	262-2	0.76	0.76
230	0.91	0.93	253	0.06	0.06	262-4	0.58	0.58
231	1	1	253-2	0.18	0.18	262-6	0.45	0.45
232	1	1	253-4	0.06	0.06	262-8	0.27	0.27
233	1	1	253-6	0.03	0.03	265	0.1	0.1
236	1	1	253-8	0.09	0.09	266	0.06	0.06
237	0.99	1	254	0.18	0.18	301	0.47	0.44
238	0.96	0.96	254-2	0.7	0.7	302	NaN	0
239	0.97	1	254-4	0.48	0.48	303	NaN	0
240	0.82	0.85	254-6	0.3	0.3	304	0.59	0.53
241	1	1	254-8	0.12	0.12			

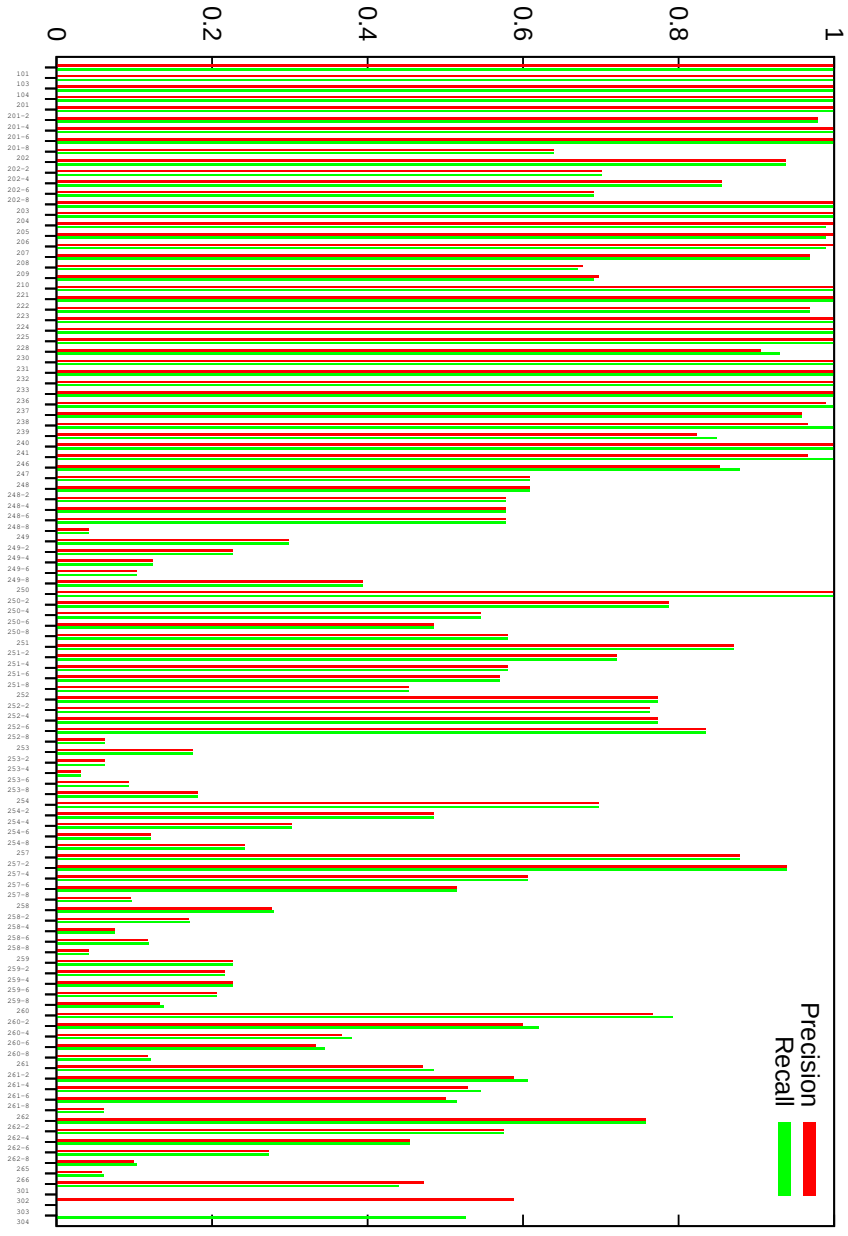


Fig. 1. Results of MapPSO in the OAEI 2009 benchmark data set.

information present in the ontologies. With respect to the quality improvement, it is planned to further investigate in the detailed implementation of these base matchers. In particular, there are plans to incorporate implicit knowledge inferred by a reasoner, as well as more sophisticated graph similarity measures. It is also necessary to review the similarity aggregation for each correspondence in order to better respect the different characteristics of different ontologies by weighting them differently.

There are further plans to deploy the system on a larger computing platform, such as a cloud infrastructure in order to utilise the full potential of the parallel nature of the system. This will be a small step with large impact, as it enables the tool to process large ontologies in reasonable time.

4 Conclusion

The results of the MapPSO system in the benchmark dataset of the OAEI 2009 have been presented. Compared to last year, the system has been extended mainly in terms of additional and refined base matchers, as proposed in the future plans section of last year's contribution [1]. This development resulted in a significant improvement of the alignment results. Future developments will focus on the scalability of the system by enabling the full potential of the parallel nature of the algorithm.

References

1. Bock, J., Hettenhausen, J.: MapPSO Results for OAEI 2008. In Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., eds.: Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008). Volume 431 of CEUR Workshop Series., CEUR-WS.org (November 2008)
2. Correa, E.S., Freitas, A.A., Johnson, C.G.: A New Discrete Particle Swarm Algorithm Applied to Attribute Selection in a Bioinformatics Data Set. In: Proceedings of the 8th Genetic and Evolutionary Computation Conference (GECCO-2006), New York, NY, USA, ACM (2006) 35–42
3. Correa, E.S., Freitas, A.A., Johnson, C.G.: Particle Swarm and Bayesian Networks Applied to Attribute Selection for Protein Functional Classification. In: Proceedings of the 9th Genetic and Evolutionary Computation Conference (GECCO-2007), New York, NY, USA, ACM (2007) 2651–2658
4. Stoilos, G., Stamou, G., Kollias, S.: A String Metric For Ontology Alignment. In Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A., eds.: Proceedings of the 4rd International Semantic Web Conference (ISWC). Volume 3729 of LNCS., Berlin, Springer (November 2005) 624–637
5. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Communications of the ACM **18**(11) (1975) 613–620

Results of OKKAM Feature Based Entity Matching Algorithm for Instance Matching Contest of OAEI 2009

Heiko Stoermer, Nataliya Rassadko

name.surname-at-unitn.it
The University of Trento
via Sommarive, 14 Povo 38123 Italy

Abstract. To investigate the problem of entity recognition, we deal with the creation of the so-called Entity Name System (ENS) which is an open, public back-bone infrastructure for the (Semantic) Web that enables the creation and systematic re-use of unique identifiers for entities. The ENS can be seen as a very large, distributed "phonebook for everything", and ENS identifiers might be considered as a "phone number" of entities. Entity descriptions are based on free-form key/value "tagging" rather than on some precise formalism. However, such a genericity has its shortcomings: the ENS can never know what type of entity it is dealing with. We tackle this problem in a novel approach for entity matching that is called Feature Based Entity Matching (FBEM). In the current paper, we report an evaluation of FBEM on datasets provided by the OAEI committee for the instance matching contest.

1 Presentation of the system

With the growth and development of Semantic Web, the latter became like a collection of "information islands" which are poorly integrated to each other. The problem of information integration in Semantic Web is two-fold:

1. heterogeneity of vocabulary: the same concept can be referred via different URIs, and therefore may be considered to be as different concepts in different vocabularies;
2. entity recognition: the same real word object can be referred via different URIs in different repositories, and therefore may not be recognized as the same object.

While the first issue is widely recognized and investigated [4], the second one was largely neglected, although it received a lot of attention under the heading of record linkage, data deduplication, entity resolution, etc [1].

To investigate the problem of entity recognition, EU-funded OKKAM project [1] deals with the creation of the so-called Entity Name System (ENS) [3].

[1] http://www.okkam.org

1.1 State, purpose, general statement

In this section, we introduce the ENS and describe our interest in instance matching part of OAEI 2009.

Entity Name System (ENS) [3] is an open, public back-bone infrastructure for the (Semantic) Web that enables the creation and systematic re-use of unique identifiers for entities. It is implemented as a large-scale infrastructural component with a set of services needed for describing entities, and assigning identifiers to them.

Figuratively, the ENS can be seen as a very large, distributed "phonebook for everything", and ENS identifiers might be considered as a "phone number" of entities. This leads to a more efficient information integration, and thus a real global knowledge space, without the need for ex-post deduplication or entity consolidation.

In the ENS, we do not impose or enforce the usage of any kind of schema or strong typing for the description of different types of entities. Instead, entity descriptions are free-form and are based on key/value "tagging". In such a way, we support a complete genericity, without the need for any formalism or any abstract top-level categorizations. Taking into account the aforementioned peculiarities of the ENS, our restriction to the instance matching part of OAEI 2009 becomes evident.

Obviously, our model of such a generic entity description has its shortcomings: the ENS can never know what type of entity it is dealing with, and how the entity is described, due to an absence of a formal model. This becomes very relevant when searching for an entity, a process that we call entity matching. To address this problem, we rely on recent work [2] that has been performed with the goal to find out in an experimental setting how people actually describe (identify) entities. Based on these findings, we propose a novel approach for entity matching.

The approach takes into account not only the similarity of entity features (keys and values), but also the circumstance that certain features are more meaningful for identifying an entity than others. We call this approach as Feature Based Entity Model (FBEM) and we explain it in the next section.

1.2 Specific techniques used

We consider both a reference (matching) entity Q and candidate (matched) entity E as a set F of *features* f:

$$F = \{f\}; f =< n, v >;$$

where each feature f is a pair of name n and value v. We do not require neither name nor value to share a vocabulary or schema, or even a natural language, i.e., they are independent in content and size.

We enumerate all features of any particular entity with integer values and denote as f_i^Q, f_j^E the ith and jth features of entities Q and E respectively.
We define the following functions:

$n(f_i)$**:** returns the *name* part of a feature of an entity;
$v(f_i)$**:** returns the *value* part.

Now, we define $f_{i,j}sim(f_Q, f_E)$, a function that computes the similarity of two features f_i^Q, f_j^E as follows:

$$f_{i,j}sim(f_Q, f_E) =_{def}$$
$$sim \quad f_i^Q, f_j^E \quad * \begin{cases} 2 * \lambda * \mu, & \text{for } name(n(f_i^Q)), name(n(f_j^E)), id(f_i^Q, f_j^E); \\ 2 * \mu, & \text{for } name(n(f_i^Q)), name(n(f_j^E)); \\ \lambda * \mu, & \text{for } name(n(f_j^E)), id(f_i^Q, f_j^E); \\ \mu, & \text{for } name(n(f_j^E)); \\ 1, & \text{otherwise .} \end{cases} \tag{1}$$

Equation 1 relies on the following functions and parameters:

$sim(x, y)$: a suitable string similarity measure between x and y.
$name(x)$: a boolean function indicating whether the feature x denotes one of the possible names of the entity;
$id(x, y)$: the identity function, returning true if value parts of x and y are identical;
μ : the factor to which a name feature is considered more important than a non-name feature;
λ : the extra-factor attributed to the the presence of the value identity $id(x, y)$.

In our implementation, we selected Levenstein metric as a similarity measure (sim-function), and both λ and μ equal to 2. The latter can be interpreted as "the occurrence of a fact is as twice as important than its absence".

We have also implemented a vocabulary, small enough to be maintained in a runtime memory, that is used to detect the cases where entity feature name is actually a "name" of the entity, e.g., "name", "label", "title", "denomination", "moniker".

At this point, we are able to establish the similarity between individual features. To compute the complete feature-based entity similarity, which finally expresses to which extend E is similar to Q, we proceed as follows.

Let $maxv(V)$ be a function that computes the maximum value in a vector[2]. We then span the matrix M of feature similarities between Q and E, defined as

$$M := (fsim\,(Q, E))_{|Q|\times|E|} \rightarrow \mathbb{Q} \geq 0$$

with $fsim$ as defined above, and $|Q|, |E|$ being the number of elements of the vectors Q and E, respectively.

The feature-based entity similarity score fs is defined as the sum of all the *maximum similar* feature combinations between Q and E:

$$fs\,(Q, E) = \sum_{i=1}^{|Q|} maxv(M_i) \tag{2}$$

[2] Trivially defined as $maxv\,(V) = max_{i=1}^{|V|}\,(V_i)$, with $|V|$ being the number of elements of V.

So far, we provided a method to calculate fs-similarity that may belong to a wide range of values from zero to infinity [5]. This complicates an evaluation of actual similarity of entities. For example, if $fs = 7$ it might stand for identical entities in one dataset and completely different entities in the other one.

To resolve this problem, we normalize fs values as follows. Taking into account that M_i is a weighted value, we use a dot-notation to denote its weight w as $M_i.w$. Then the final formula of *normalized* similarity has the following form:

$$esim(Q, E) = \frac{fs(Q, E)}{\sum_{i=1}^{|Q|} maxv(M_i).w} \quad (3)$$

In the last formula, we simply divided a sum of weighted values on a sum of corresponding weights. This allows us to normalize similarity score within the range of $[sim(x, y)_{min}, sim(x, y)_{max}]$, e.g., $[0, 1]$ if similarity metric return the values in this range, which is true for Levenstein similarity.

1.3 Adaptations made for the evaluation

We parsed all provided rdf-files into a Jena-model [3] stored as a persistent SDB [4] with an underlying MySQL database [5]. To adapt our FBEM-model to the required output in the alignment format [6], we wrote a simple iterator over SDB-instances related to reference entities Q and to candidate entities E, i.e., we matched each Q against each E, where both Q and E were preliminarily converted to the ENS entity format.

For the reason of a better time-performance, we implemented a "typed" matching, i.e., Q and E should have been of the same entity type (e.g., people were matched against people, documents against documents). The types were easy to extract from the attribute "type" available in most benchmarks. We also implemented a "brute-force" matching, i.e. any-to-any, which did not consider any type features, to match those benchmarks where typing was not provided or was difficult to reason.

For each Q, we maintained a vector of E ranked w.r.t. a similarity value $esim(Q, E)$. The length of vector was limited to 50 elements due to time- and memory- performance reasons.

In the alignment file, we output only those elements of vector of Es that had a similarity value greater than or equal to a certain threshold. The threshold was selected empirically for each particular benchmark. More precisely, we run experiments for thresholds from the set $\{0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95\}$ and then selected that thresholds that gave us the most acceptable values of precision/recall from the viewpoint of the ENS methodology. Namely, we were eager to maintain as high precision as possible with any non-zero recall.

The reason for selecting precision of the ENS performance was the following: we assume that the ENS user, while querying the ENS repository, expects few answers in

[3] http://jena.sourceforge.net/

[4] http://jena.sourceforge.net/SDB/

[5] http://mysql.com

[6] http://alignapi.gforge.inria.fr/format.html

the result set. However, these answers should be the most relevant to the user query. In other words, for the ENS it's better to answer with some highly precise entities rather than with a lot of somehow likely similar entities.

Precise threshold values we used to run FBEM-matching over each particular benchmark will be indicated in Sec. 2.

1.4 Link to the system and parameters file

`http://www.dit.unitn.it/~rassadko/OAEI2009/okkamsystem.zip`

1.5 Link to the set of provided alignments (in align format)

`http://www.dit.unitn.it/~rassadko/OAEI2009/okkamalignment.zip`

2 Results

Due to peculiarities of the ENS described in Sec. 1.1, we have restricted ourselves only to instance matching benchmarks.

2.1 A-R-S

The benchmark contains includes three datasets describimg instances from the domain of scientific publications:

- eprints - this dataset contains papers produced within the AKT research project and extracted using an HTML-wrapper from the source web-site;
- rexa - this dataset was extracted from the search results of the search server;
- SWETO-DBLP - a version of the DBLP dataset.

For A-R-S benchmark we applied a "typed" version (see Sec. 1.3) of FBEM-matching because all three datasets contained information about authors (typed with *foaf* namespace [7]) and their scientific publication (typed with *opus* namespace [8]).

We run our experiment with threshold 0.80. The result of our experiments are presented in Table 1.

In Sec. 1.3, we explained that we are interested in high precision with any non-zero recall. As Table 1 shows, we gained our objective. With a less tight threshold, it is possible to slightly sacrifice a precision for a better recall.

2.2 T-S-D

For this dataset we do not have results. First of all, typing of each particular data source was different from the others. This required reasoning over ontologies which were provided with datasets. Since our system does not support any kind of ontology reasoning, one might have made an attempt to run a "brute-force" matching, i.e., any-to-any. Unfortunately, due to a large size of data, we were unable to finish the match run timely.

[7] `http://xmlns.com/foaf/0.1/`
[8] `http://lsdis.cs.uga.edu/projects/semdis/opus`

Table 1. A-R-S results

Test	Precision	Recall	F-measure	Fallout
eprints-rexa	0.94	0.10	0.18	0.06
eprints-dblp	0.98	0.16	0.28	0.02
rexa-dblp	1.00	0.12	0.22	0.00

2.3 IIMB

IIMB benchmark is generated from a dataset provided by OKKAM. We run our experiment with threshold 0.95. Our results are shown in Table 2.

Table 2. IIMB results

Test	001	002	003	004	005	006	007	008	009	010
Precision	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.95	0.92
Recall	1.00	1.00	1.00	1.00	0.99	0.98	0.98	0.96	0.85	0.52
F-measure	0.98	0.98	0.98	0.98	0.97	0.97	0.97	0.96	0.90	0.66
Test	011	012	013	014	015	016	017	018	019	
Precision	0.88	0.94	0.92	0.00	0.91	0.86	0.72	0.82	0.71	
Recall	0.43	0.98	0.71	0.00	0.96	0.74	0.30	0.38	0.15	
F-measure	0.58	0.96	0.80	NaN	0.93	0.79	0.43	0.52	0.25	
Test	020	021	022	023	024	025	026	027	028	029
Precision	0.78	0.47	0.15	0.08	0.09	0.05	0.09	0.05	0.89	0.00
Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	NaN
F-measure	0.88	0.64	0.25	0.15	0.17	0.10	0.16	0.10	0.94	NaN
Test	030	031	032	033	034	035	036	037		
Precision	0.28	0.80	0.09	0.08	0.10	0.00	0.11	0.00		
Recall	0.04	0.25	1.00	0.99	1.00	0.00	0.98	0.00		
F-measure	0.06	0.38	0.16	0.15	0.19	NaN	0.19	NaN		

Below, we provide our comments to the results presented in Table 2:

001 Contains an identical copy of the original ABox with the instance IDs randomly changed. And for this test, we performed well with pretty high precision.

002-010 Value transformations (i.e., typographical errors simulation). ENS user is not assumed to enter extremely misspelled queries. Therefore, we may conclude that our performance is appropriate. Although the recall dropped down at experiment 010, ENS user would still received highly relevant result set.

011-019 Structural transformations (i.e., deletion of one or more values, transformation of datatype properties into object properties, separation of a single property into more properties). From ENS viewpoint it might be seen as if the user query contained permutated feature names and feature values. For these test cases, we

have medium performance: with the precision around 0.70-0.90, the recall varies from 0.15 to 0.98. We believe, that these results are still acceptable for the ENS user.

020-029 Logical transformations (i.e., instantiation of identical individuals into different subclasses of the same class, instantiation of identical individuals into disjoint classes, instantiation of identical individuals into different classes of an explicitly declared class hierarchy). These cases are impossible for ENS because ENS does not have any schema or ontology. Yet having conducted a "brute-force" (non-typed) matching of each entity Q against each entity E, we could still provide the ENS user with some information.

030-037 Several combinations of the previous transformations. For these test cases, we have an uneven performance which is expected.

3 General comments

3.1 Comments on the results

We mainly commented our results in Sec. 2. In general, we believe that FBEM performs well for the purposes of the ENS. Namely, we are able to answer user queries with a high precision. And this is a strength of our approach. As the weakness, we have to admit that recall values are not so much satisfactory. And in the next section, we will discuss the ways to deal with this problem.

3.2 Discussions on the way to improve the proposed system

We need to experiment with other similarity metrics $sim(x, y)$ since Levenstein metrics deals badly with the permutated words, e.g., "Stephen Potter" and "Potter, Stephen". This can lead to a low recall as in our results for A-R-S benchmark.

Basic structural analysis is also planned to be introduced. For example, one entity Q may have attributes "first name" and "given name" while entity E can contain only "name" (i.e. both first and give name together). We believe that elements of structural analysis will help us improve both precision and recall for the cases like in tests 20-29 for IIMB benchmark.

We are currently working on a more extended version of FBEM-model which concentrates not only on names of entities, but also on other features that might identify entity. For example, a feature "isbn" uniquely identifies book, "e-mail" likely identifies a person etc. We will rely on the empirical study [2] which we mentioned above.

Finally, we did not expect the datasets larger than 1Gb. However, this forced us to include in our future research also a loaded bulk-matching, e.g., 1Gb dataset against 1Gb dataset.

3.3 Comments on the OAEI 2009 procedure

We are satisfied with the OAEI 2009 procedure.

3.4 Comments on the OAEI 2009 test cases

As we said above, the test cases turned to be unfeasible for our matching procedure.

3.5 Comments on the OAEI 2009 measures

We are satisfied with the OAEI 2009 measures.

3.6 Proposed new measures

No proposals.

4 Conclusion

In the current paper, we proposed an evaluation of a novel approach for entity matching that is called Feature Based Entity Matching (FBEM) over datasets provided by the OAEI committee for the instance matching contest.

Since FBEM could be a candidate to a set of matching modules of the ENS, we were eager to maintain as high precision as possible with any non-zero recall. In general, we gained our objective. Namely, we perform well in the cases where there is no need in ontology reasoning or structural analysis.

We are satisfied with our results. However, there are several directions (see Sec. 3.2) to improve the performance of FBEM from the viewpoint of both precision and recall values.

Acknowledgments. This paper has been supported by the FP7 EU Large-scale Integrating Project OKKAM "Enabling a Web of Entities" (contract no. ICT-215032). For more details, visit `http://fp7.okkam.org`.

References

1. Duplicate record detection: A survey. *IEEE Trans. on Knowl. and Data Eng.*, 19(1):1–16, 2007. Senior Member-Elmagarmid, Ahmed K. and Member-Ipeirotis, Panagiotis G. and Member-Verykios, Vassilios S.
2. B. Bazzanella, P. Bouquet, and H. Stoermer. A Cognitive Contribution to Entity Representation and Matching. Technical Report DISI-09-004, Ingegneria e Scienza dell'Informazione, University of Trento., 2009. `http://eprints.biblio.unitn.it/archive/00001540/`.
3. P. Bouquet, H. Stoermer, C. Niederee, and A. Mana. Entity Name System: The Backbone of an Open and Scalable Web of Data. In *Proceedings of the IEEE International Conference on Semantic Computing, ICSC 2008*, CSS-ICSC, pages 554–561. IEEE Computer Society, 2008.
4. J. Euzenat and P. Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.
5. H. Stoermer and P. Bouquet. A Novel Approach for Entity Linkage. In *Proceedings of IRI 2009, the 10th IEEE Internationational Conference on Information Reuse and Integration, August 10-12, 2009, Las Vegas, USA*, volume 10 of *IRI*, pages 151–156. IEEE Systems, Man and Cybernetics Society, August 2009.

RiMOM Results for OAEI 2009

Xiao Zhang, Qian Zhong, Feng Shi, Juanzi Li and Jie Tang

Department of Computer Science and Technology, Tsinghua University, Beijing, China
zhangxiao,zhongqian,shifeng,ljz,tangjie@keg.cs.tsinghua.edu.cn

Abstract. In this report, we give a brief explanation of how RiMOM obtains the results at OAEI 2009 Campaign, especially in the new Instance Matching track. At first, we show the basic alignment process of RiMOM and different alignment strategies in RiMOM. Then we give new features in instance matching compared with traditional ontology matching (schema matching) and introduce the specific techniques we used for the 3 different subtracks of Instance Matching Track. At last we give some comments on our results and discuss some future work about RiMOM.

1 Presentation of the system

Ontology matching is the key technology to reach interoperability over ontologies. In recent years, much research work has been conducted for finding the alignment of ontologies[1]. Many automatic matching algorithms achieves good results in real world data. With the development of Linked Data[2], huge amount of semantic data are available through the web. Thus instance matching, a special branch of ontology matching, draws lots of research interest recent years.

RiMOM is a multiple strategy dynamic ontology matching system implemented in Java [3]. In RiMOM, we implement several different matching strategies. Each strategy is defined based on one kind of ontological information. Moreover, we investigate the differences between the strategies and compare the performances of different strategies on different matching tasks. We propose a mechanism in RiMOM to choose appropriate strategies (or strategy combination) according to the features and the information of the ontologies. RiMOM can deal with unbalanced ontology matching [4]. We also try to bring user interaction into RIMOM [5]. This year We modified the RiMOM system to make it with better support for instance matching.

1.1 State, purpose, general statement

RiMOM is a framework for ontology matching. Different kinds of alignment strategies can be added into RiMOM. Based on the features of the input ontology and the defined rules, appropriate strategies are chosen to apply for the matching task. The basic matching process of RiMOM is shown in figure 1.

There are six major steps in a general alignment process of RiMOM.

– Ontology Preprocessing and Feature Factors Estimation. The input ontologies are loaded into the memory and the ontology graph is constructed. Some redundant or useless information are removed. Then the three ontology feature factors used in strategy selection are estimated.

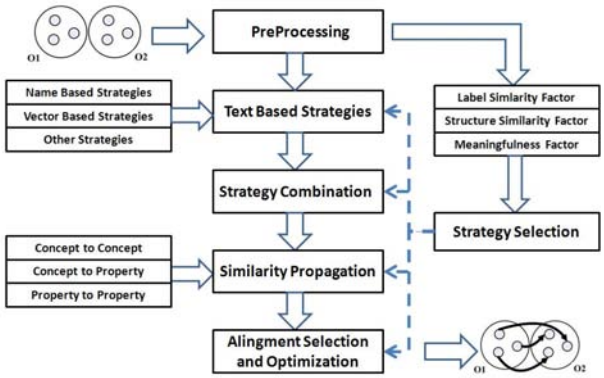


Fig. 1. The Alignment Process of RiMOM

- Strategy Selection. The basic idea of strategy selection is that if two ontologies have some same feature, then strategies based on these feature information are employed with high weight; and if some feature factors are two low, then these strategies may be not employed. For example, the string based strategy will be used when the label Similarity factor is high while the WordNet [6] based stratey will not be used when the label meaningful factor is low.
- Single strategy execution. We get the selected strategies to find the alignment independently. Each strategy outputs an alignment result.
- Alignment combination. In this phase RiMOM combines the alignment results obtained by the selected strategies. The combination is conducted by a linear-interpolation method.
- Similarity propagation(Optional). If the two ontologies have high structure similarity factor, RiMOM employs a similarity propagation process to refine the found alignment and to find new alignment according to the structural information.
- Alignment refinement. It refines the alignment results from the previous steps. We defined several heuristic rules to remove the ”unreliable” alignments.

1.2 Specific techniques used

This year we participate four tracks of the campaign: Benchmark, Anatomy, Oriented Matching and Instance Matching. The Benchmark and Anatomy dataset is almost the same as last year. The Oriented Matching dataset is very similar to the Benchmark one. We focused on the new and challenging Instance Matching Track.

Benchmark and Anatomy Track The strategy we use for Benchmark and Anatomy track is almost the same for OAEI 2008, more detailed explanation of the strategies used could be found in [7] [8].

Oriented Matching Track The dataset of oriented matching track is derived from the benchmark dataset. Naturally, we combined the methods we use in the benchmark track and the sub relation in the ontology graph. Since RiMOM's performance for the Benchmark Track is fairly good, the result shows that the combination also works for the Oriented Matching Track. This technique is also applied for the schema matching phase of the Instance Matching Track.

Instance Matching Track The Instance Matching Track is introduced into the campaign this year. The traditional ontology matching focus on the schema matching and the ontology may contain no individual. If there are small amount of individuals, the alignment of individuals are usually used to enhance the alignment of concepts and properties. Previous OAEI campaigns also evaluated the performance of Matching Systems according to the schema matching results. By analyzing the datasets, we found some differences between the traditional ontology matching and instance matching. We summarize the differences as following:

- Ontology is used as a formal, explicit specification of a shared conceptualization[9]. It defines the concepts of the domain and the relation between the concepts. That is to say, it describe the domain in the concept layer. However, the instance is the instantiation of the ontology, it is composed of concrete values of the domain, and has rich practical semantic information. As we observe, some attribute values may clearly different from others. How to find the key attributes and key values of instances to facilitate the process of ontology matching is a very challenging issue.
- The ontology can be viewed as a whole ontology graph and some graph based algorithms are employed in ontology matching. However, a concept may have lots of instances and all the instances are with almost the same structure. The graph algorithm with the whole ontology graph is not suitable for the instance matching task. For a given instance, some other instances related to it may contain information about it through object properties. Moreover, the information in the instance may be not symmetric as in the ontology. For instance, an instance A of the "Author" concept is "list_author_in" an instance B of the "Scientific_Publication" concept. However, the statement is only written in B's description. How to find the complete information of a given instance is also very important.
- The scale of instance matching is usually much more larger than ontology matching. The sweto data and the dbpedia data file both contain more than 800,000 instances which seems impossible in ontology matching. As a result, the efficiency of the instance matching strategies becomes a major concern. Some complicate algorithm can not be employed.
- The schema ontology for most instance files are available. So the result of ontology matching is a very good background knowledge for instance matching. Instinctively, the instance pair of not matched concept pair have no chance to be matched. So with the ontology alignment, the number of instance matching candidates can be pruned greatly. However, sometimes the ontology itself is not very well defined. For example, The DBpedia ontology does not cover all the instances in the instance file, so the ontology itself should be enriched with instance type information.

With regard to the different characteristics of the three different subtracks of Instance Matching (We do not take part in the vlcr subtrack), we employ some different strategies to solve the three tasks.

The A-R-S benchmark includes three datasets containing instances from the domain of scientific publications. The three data file are quite different in size, especially the DBLP file is really large. So we use some light-weight method in this subtrack. On the other hand, all the three data files are mainly in the scientific publication domain. At first, we choose some data-type properties as the key attributes carefully. These properties are of two types. The first type is the "sufficient" property group: if the values of the properties between two instances are matched respectively, the two instances are considered as matched. The second type is the "necessary" property group: if two instances are marked as matched, the corresponding values should be matched. The "sufficeint" properties, such as foaf:name are employed in a edit-distance strategy to find the initial alignment. The "necessary" properties, such as the opus:year, are employed to refine the initial alignment. In the second step, a structure based matching method is used to propagate the similarities among the instances according to the object properties. For example, we can refine the "person" matches with the "document" alignments in terms of the opus:author property.

Compared with the A-R-S benchmark which is restricted to the scientific publication domain, the T-S-D benchmark covers much more wider domains. The DBpedia data is encyclopedia-like knowledge base. This makes it difficult to find particular attributes and values, so we take another strategy. First of all, we compute the schema matching results with RiMOM, check the incorrect alignments and add the missing ones carefully. Then for every instance, we generate a vector to describe the information contained in the instance. The vector contains labels of the instance, data-type property values of the instance, labels and property values of the instances related to the underlying one through object-type properties. Then the similarity of the instance pair of matched concept pair is calculated by a vector based algorithm. The weight of respective element of the vector is designated by heuristic rules defined based on the structure of the instance. The instance pairs of non-matched concept pairs are discarded directly. In the DBpedia data file, there are some instances missing the rdf:type information. We try to match these instances with all source instances in the reference file.

The IIMB benchmark, on the other hand, is systematically generated. The dataset is constituted using one data file by modifying it according to various criteria. The variation can be sorted into three types: value transformation, structural transformation and logical transformation. The purpose of value transformation is to simulate typographical errors, so edit-distance strategy employed on the relevant property values between instances is effective enough. In structural transformation, some data-type properties may transformed in the form of object-type property. We design an property value passing approach to cope with this kind of modification. The data-type property value of the instance are passed to the instances connected with it through a object-type property. We also consider structural information when matching instances. If two instances have more property values on the same properties, they will be considered more similar. In logical transformation, the TBox is modified, so we matching the TBox first to find the find the relations between concepts in the TBoxes, then we try to match instances

according to the type information. In addition, some instance pairs with very similar property values but with non-matched concepts are checked. If they can match each other, then we consider their concepts are matched to enhance the TBox alignment. When the two-direction matching process convergence to a stable matching result, we take it as the final output.

1.3 Adaptations made for the evaluation

To reduce the number of matching candidate in T-S-D benchmark subtrack of Instance Matching Track, the schema matching alignments is refined manually by correcting some incorrect alignments and adding missing ones.

1.4 Link to the system and parameters file

The RiMOM System can be found at `http://keg.cs.tsinghua.edu.cn/project/RiMOM/`

1.5 Link to the set of provided alignments (in align format)

The results for OAEI 2009 Campaign are available at `http://keg.cs.tsinghua.edu.cn/project/RiMOM/OAEI2009/`

2 Results

As mentioned above, RiMOM takes part in four tracks of campaign in OAEI 2009. Normally RiMOM uses OWL-API[10] to parse RDF and OWL Files. RiMOM also uses Jena API[11] to convert N3 format files into RDF files and to deal with some large scale instance files. The Benchmark, Oriented Matching and IIMB matching tasks are carried out on a PC running Window XP with AMD Athlon 64 X2 4200+ processor(2.19GHz) and 2GB memory. To run the large scale matching tasks, Anatomy, A-R-S benchmark and T-S-D benchmark, the experiments are carried out on a server running Ubuntu Server 8.10 with two 4-core Intel Xeon E5440 processors(2.83GHz) and 32GB memory.

2.1 Benchmark

There are in total 111 alignment tasks defined on the benchmark data set. RiMOM takes exactly the general process of matching. However, on the tasks where the labels are absolutely random strings, the WordNet based strategy and edit-distance based strategy are not used. The vector-similarity based strategy is always employed. RiMOM maintains the high performance on benchmark as previous years.

2.2 Anatomy

The anatomy data set contains two large scale anatomy ontologies. RiMOM first extract the labels of all concepts from rdfs:label and oboInOwl:Synonym property. The match process first employs edit-distance based strategy on labels to get the initial mapping, then RiMOM propagates the similarity on both the concept hierarchy ant the object property "UNDEFINED_part_of" to get the alignments which cannot be extracted by just comparing the labels simply. Since the structure of the two ontologies is somehow not that similar, we restricted the propagation for every concept locally.

2.3 Oriented Matching

Because our strategy in oriented matching is the combination of the strategy in the Benchmark dataset and structure based stratedy by using the rdfs:subclass property. The result relies heavily on the Benchmark strategy and shows the same characteristics as in the Benchmark dataset. Except in the files the name of the entities are totally random string and nearly no other information are available, RiMOM achieves satisfying results.

2.4 Instance Matching

The result for A-R-S benchmark is as Table 1 shows. RiMOM produces alignments all with an F-Measure in the range of about 0.75+. The result relate to eprints data have both high precision than the rexa-dblp one.

Table 1. Result of A-R-S Benchmark

Data	Precision	Recall	F-Measure
eprints-rexa	0.928	0.699	0.797
eprints-dblp	0.930	0.671	0.780
rexa-dblp	0.805	0.725	0.763

The T-S-D benchmark is a blind test, so we do not know the final results for it now. According to our observation on our alignment, about 30% to 50% of the instances in the reference files are matched. It seems indeed that most of instances in the reference can not find a correspondence. Since we choose a relatively high threshold in the final alignment extraction, we believe the result is of high precision.

Except the dataset 028 which seems missing some correct alignments and the dataset 029 which do not contain a reference alignment, the result for IIMB benchmark is as Table 2 shows. We can see that RiMOM can achieve perfect alignment in more than half of the dataset. Only for the dataset 017 in which the information is severely suppressed, RiMOM can only get an F-Measure less than 0.90. RiMOM is quite successful in IIMB dataset.

Table 2. Result of A-R-S Benchmark

Dataset	Precision	Recall	F-Measure
001 - 014	1.0	1.0	1.0
015	1.0	0.991	0.995
016	1.0	0.910	0.953
017	0.993	0.626	0.768
018	1.0	0.986	0.993
019	1.0	0.883	0.938
020 - 027	1.0	1.0	1.0
030	1.0	1.0	1.0
031	1.0	0.892	0.943
032 - 037	1.0	1.0	1.0

3 General comments

Except for performing the ontology matching tasks like the previous years, this year we concentrate on the new and interesting Instance Matching Track. We first modify the infrastructure of RiMOM to make it to support instance matching naturally. The results shows that now RiMOM can handle instance matching tasks with good performance. But there are still many future works to do:

- Although instance matching is regarded as a subtask of ontology matching, the model of instance matching is different from traditional schema matching to some extent. Some algorithms in schema matching can not be imported into instance matching directly. In addition, instance matching seems more close to practical use than schema matching. This makes it a very attractive research topic.
- The scalability problem is very critical in instance matching. The scale of instance files are greatly larger than the schema files and the execution times and memory needs grows very fast as the input scale increases. For example, our strategy for A-R-S benchmark consumes more than 36 hours to generate the alignment on our 8-core server while the strategy itself is not that complicated. How to solve this problem is a big challenge. We may try to introduce the database-like techniques into RiMOM to make it support the large scale instance data better.
- Because the instance data are retrieved from the real web data, so it usually contains more semantic information than the theoretically designed schemas. However, most of our approaches are string based comparisons and so on. How to dig the deeperthe semantics in the instance is another work.

4 Conclusion

In this report, we give aa briefly explanation of how we employed RiMOM to obtain the alignment results in OAEI 2009 Campaign. We have presented the alignment process of RiMOM and explained the strategy defined in RiMOM. We focus on the Instance Matching Track, analyzing the feature of instance matching and introduce the strategies

we use in this track. The experiments illustrates that our system RiMOM can achieve good results in both schema matching and instance matching tracks. We also discuss the future work we will do to improve our system.

Acknowledgement

The work is supported by the National Natural Science Foundation of China (No. 60973102 and No. 60703059), the National Basic Research Program of China (973 Program) (No. 2007CB310803), the National High-tech R&D Program (No. 2009AA01Z138), and the Chinese Young Faculty Research Fund (No. 20070003093).It is also supported by IBM SUR joint project.

References

1. J. Euzenat and P. Shivako. *Ontology Matching*. Springer-Verlag, Berlin Heidelberg (DE), 2007.
2. `http://linkeddata.org/`.
3. J. Li, J. Tang, Y. Li, and Q. Luo. RiMOM: A dynamic multi-strategy ontology alignment framework. *IEEE Transaction on Knowledge and Data Engineering*, 21(8):1218–1232, Aug 2009.
4. Q. Zhong, H. Li, J. Li, G. Xie, and J. Tang. A Gauss Function based approach for unbalanced ontology matching. In *Proc. of the 2009 ACM SIGMOD international conference on Management of data (SIGMOD'2009)*, Jul 2009.
5. F. Shi, J. Li, and J. Tang. Actively learning ontology matching via user interaction. In *Proc. of the 8th International Conference of Semantic Web (ISWC'2009)*, Oct 2009.
6. `http://wordnet.princeton.edu/`.
7. Y. Li, J. Li, D. Zhang, and J. Tang. Results of ontology alignment with RiMOM. In *Proc. of the Second International Workshop on Ontology Matching (OM'07)*, Nov 2007.
8. X. Zhang, Q. Zhong, J. Li, J. Tang, G. Xie, and H. Li. RiMOM results for OAEI 2008. In *Proc. of the Third International Workshop on Ontology Matching (OM'08)*, 2008.
9. T. R. Grubber. A translation approach to portable ontology specification. *Knowledge Acquisition*, 5:199–200, 1993.
10. `http://owlapi.sourceforge.net/`.
11. `http://jena.sourceforge.net/`.

Alignment Results of SOBOM for OAEI 2009

Peigang Xu, Haijun Tao, Tianyi Zang, Yadong, Wang

School of Computer Science and Technology

Harbin Institute of Technology, Harbin, China

xpg0312@hotmail.com, hjtao.hit@gmail.com, tianyi.zang@gmail.com, ydwang@hit.edu.cn

Abstract. In this paper we give a brief explanation of how Anchor Concept and Sub-Ontology based Ontology Matching (SOBOM) gets the alignment results at OAEI2009. SOBOM deal with the ontology from two different views: an ontology with is-a hierarchical structure O' and an ontology with other relationships O''. Firstly, from the O' view, SOBOM starts with a set of anchor concepts provided by linguistic matcher. And then it extracts sub-ontologies based on the anchor concepts and ranks these sub-ontologies according to their depth. Secondly, SOBOM utilizes Semantic Inductive Similarity Flooding algorithm to compute the similarity of the concepts between the sub-ontologies derived from the two ontologies according the depth of sub-ontologies to get concept alignments. Finally, from the O'' view, SOBOM gets relationship alignments by using the concept alignment results in O''. The experiment results show SOBOM can find more alignment results than other compared relevant methods with high degree of precision.

1 System presentation

Currently more and more ontologies are distributedly built and used by different organizations. And these ontologies are usually light-weighted [1] containing lots of concepts especially in biomedicine, such as anatomy taxonomy NCI thesaurus. The Anchor Concept and Sub-ontology based Ontology Matching (SOBOM) is designed for matching light-weight ontologies. It handles an ontology from two views: O'

and O'' that are depicted in Fig. 1. The unique feature of our method is combining sub-ontology extraction with ontology matching.

1.1 State, purpose, general statement

SOBOM is an automatic ontology matching tool. There are three matchers implemented in current version: linguistic matcher I-Sub [2], structure matcher SISF (Semantic Inductive Similarity Flooding) which was inspired by Anchor-Prompt [3] and SF [4] algorithms, and relationship matcher R-matcher which utilizes the results of SISF to get relationship alignments. In addition, a Sub-ontology Extractor (SoE) is integrated into SOBOM to extract sub-ontologies according to the result of I-Sub and rank them. The method of SOBOM is fully sequential, so it does not care how to combine the results of different matchers. The overview of the approach is illustrated in Fig. 2.

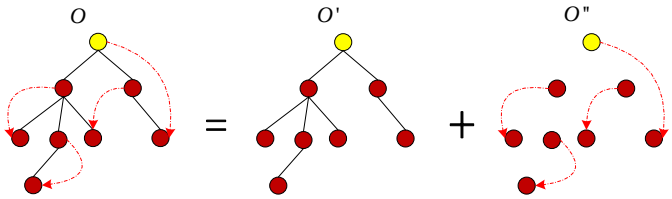


Fig. 1. The construction of ontology in SOBOM

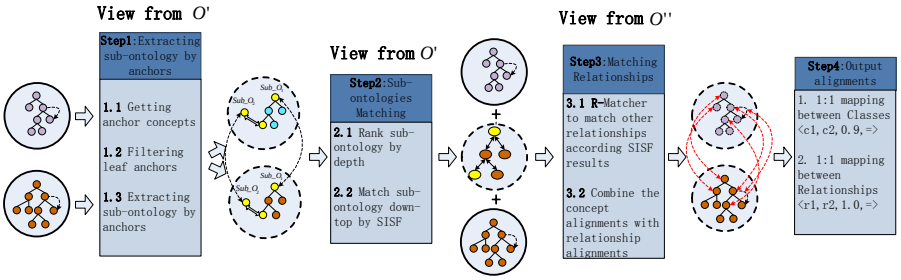


Fig. 2. The process of SOBOM algorithm

For simplicity, we define some notations used in the report.

Ontology: An ontology O consists of a set of concepts C, properties/relations R, instances I, and Axioms A^O. We use entity e to denote either $c \in C$ or $r \in R$. Each relation r has a domain and range defined as following:

$$Domain(r) = \{c_i \mid c_i \in C \text{ and having the relationship } r\}$$

$$Range(r) = \{c_i \mid c_i \in C \text{ and can be value of } r\}$$

Anchor concept: an anchor concept is the strongest semantic similarity between two entities in different ontologies. It is a pair of concepts from two ontology:

$a = < c_1, c_2 >$, where $sim(c_1, c_2) \geq \mu$

Sub-Ontology: Sub_O is a concept hierarchy with $a.c_i$ as root, it satisfied that $\forall c \in Sub\_O$, $c \in O$ and c is a descendant of $a.c_i$, $Sub\_O \subseteq O$。

1.2 Specific techniques used

SOBOM aims to provide high quality of 1:1 alignments between concept and property pairs. We implemented SOBOM algorithm in java and had integrated three distinguishing constitutional matchers, I-Sub, SISF and R-matcher. They are regarded as independent components in core matcher library of SOBOM. Due to the space limitation, we only describe the key features of them. The details can be found in the related paper [8].

- I-Sub is a light-weighted matcher simply based on the string comparison techniques. The innovation of I-Sub is not only the commonalities between the descriptions of domain entities are calculated but also their differences are examined. Furthermore, it is stable to small diverges from the optimal threshold taking place and intelligent to identify all the differences between strings. In SOBOM, I-Sub is a core component to generate anchor concepts.
- SISF uses the RDF statement to represent the ontology and utilizes the results of I-Sub to inducting the construction of similarity propagation graph from

sub-ontologies. SISF and I-Sub handle the ontology from the view O' and only generate concept-concept alignment.

- R-matcher is a relationship matcher base on the definition of the ontology. It combines the linguistic and semantic information of a relation. From the O'' view, it utilizes the is-a hierarchy to extend the domain and range of a relationship and uses the result of SISF to generate the alignment between relationships.

More importantly, SoE is integrated into the SOBOM and extract sub-ontologies according to the anchor concept [5, 6]. SoE ranks extracted sub-ontologies from the O' view according to their depth. As for ontology matching, the rules of extracting sub-ontology in SoE are as following:

Rule 1: Upwards traversal of the hierarchy: $\forall c' \in O$, if c' is an ancestor of $a.c_i$, then $c' \notin Sub\_O$.

Rule 2: Siblings classes of anchor concepts: $\forall c' \in O$, if c' is a sibling concept of $a.c_i$, then $c' \notin Sub\_O$.

Rule 3: Downwards traversal of the hierarchy: $\forall c' \in O$, if c' is descendant concepts of $a.c_i$, $c' \in Sub\_O$

Rule 4: Other relationships of the anchor concepts: $\forall r \in O$, if r is a relationship in O and $r \neq is\_a$, then $r \notin Sub\_O$

Rule 5: Leaf Concept Nodes: if $a.c_1 \in O_1$, $a.c_2 \in O_2$ and $a.c_1$, $a.c_2$ are leaf nodes respectively in O_1, O_2, then don't extract Sub-Ontology.

After extracting sub-ontologies, SOBOM will match these sub-ontologies according to their depth in original ontology. We first match the sub-ontologies with larger depth value. By using SoE, SOBOM can reduce the scale of ontology and make it easy to operate sub-ontologies in SISF.

1.3 Adaptations made for the evaluation

We don't make any specific adaptation for the tests in the OAEI 2009 campaign. All the alignments outputted by SOBOM are based on the same set of parameters.

1.4 Link to the system and set of provided alignments (in align format)

The current version of SOBOM and the alignment results for OAEI 2009 are available at http://mlg.hit.edu.cn:8080/Ontology/Download.jsp, and the parameters setting is illustrated in the reading me file.

2 Results

In this section, we describe the results of SOBOM algorithm against the benchmark, directory and anatomy ontologies provided by the OAEI 2009 campaign. We use Jena-API to parse the RDF and OWL files. The experiments were carried out on a PC running Windows vista ultimate (32 bit) with Core 2 Duo processors (2.66 GHz) and 4-gigabyte memory.

2.1 Benchmark

On the basis of the nature, we can divide the benchmark dataset into five groups: #101-104, #201-210, #221-247, #248-266 and #301-304. We described the performance of our SOBOM algorithm over each group and overall performance on the benchmark test set in Table 1.

#101-104 SOBOM plays well for these test cases.

#201-210 In this group, some linguistic features of candidate ontologies are discarded or modified. SOBOM is a sequential matcher, if the linguistic matcher get no mappings, then the SISF will produce no mapping too. So in these test, the result is in high precision but low recall.

#221-247 The structures of the candidate ontologies are altered in these tests. However, SOBOM discovers most of the alignments from the linguistic perspective via our linguistic matcher, and both the precision and recall are pretty good.

#248-266 Both the linguistic and structural characteristics of the candidate ontologies are changed heavily. In most cases, SOBOM can get high precision but low recall.

#301-304 This test group are four real-life ontologies of bibliographic references. SOBOM can only find equivalence alignment relations.

Table 1. The performance on the benchmark

	101-104	201-210	221-247	248-266	301-304	Average	H-mean
Precision	0.98	0.99	0.99	1.0	0.86	0.96	0.98
Recall	0.97	0.48	0.95	0.43	0.52	0.67	0.43

2.2 Anatomy

The anatomy real world test bed covers the domain of body anatomy and consist of two ontologies, Adult Mouse Anatomy (2247 classes) and NCI Thesaurus (3304 classes). This type ontologies is what SOBOM suitable for. The experiment result shows in Table 2.

Table 2. The performance of SOBOM on the anatomy test

	Anchor-concept	Sub-ontologies	Alignments	Time consuming
NCI	1233	268	1249	19min3s
MA				

2.3 Directory

The directory track requires matching two taxonomies describing the web directories. It includes 4639 matching tasks represented by pairs of OWL ontologies, where classification relations are modeled as *rdfs:subClassOf* relations. But in the

experiments, we found there are some ontologies have wrong structure, they have a loop such as 1603, 1704, 2114, 2184, 2241, 2252, 2416, 3045, 3135, 3166, 3183, 3301,3398, 3440, 3556, 3653, 3695, 3711, 4075, 4129, 4544, 851, 118, 148, 1550,1723, 1863, 1967,2, 2000, 2103, 2270, 2271, 2632, 2749, 2803, 3058, 3186,3310, 3455, 3461, 3891, 4048, 4089, 4116, 4341, 4556, 614, 726, 747, totally 50 ontologies. So SOBOM cannot deal with these tests. The experiment results shows in Table 3.

Table 3 The performance of on directory test

Precision	Recall	F-measure
0.5931	0.4145	0.4879

3 General comments

3.1 Comments on the results

Strengths SOBOM deals with ontology from two different views and combines results of every step in sequential way. If the ontologies have regular literals and hierarchical structures, SOBOM can achieve satisfactory alignments. And it can avoid missing alignment in many block matching methods [7].

Weaknesses SOBOM needs the anchor concepts to extract sub-ontologies. So it heavily depends on the anchor concepts. if the literals of concept missed, SOBOM will get bad results.

3.2 Discussions on the way to improve the proposed system

SOBOM can be viewed as a frame of ontology matching. So many independent matchers can be integrated into it. Now anchor concepts generator is a weak matcher, our next plan is to integrate a more powerful matcher to produce anchor concepts or develop a new method to get anchor concepts.

4 Conclusion

This paper reports our first participation in OAEI campaign. We present the alignment process of SOBOM and describe the specific techniques for ontology matching. We also show the performance in different alignment tasks. The strengths and the weaknesses of our proposed approach are summarized and the possible improvement will be made for the system in the future. We propose a brand new algorithm to match ontologies.

References

1. Fausto Giunchiglia and Ilya Aihrayeu : Lightweight Ontologies. Technical Report. (2007) DIT-07-071.
2. G.Stoilos, G. Stamou, S. Kollias: A string metric for ontology alignment, In Proc. Of the 4th International Semantic Web Conference(ISWC'05). (2005) 623-637.
3. N.F. Noy, M.A. Musen: Anchor-PROMPT: using non-local context for semantic matching, In Proc. Of IJCAI2001 Workshop on Ontology and Information Sharing, (2001) 63-70.
4. S. Melnik, H.G. Molina and E. Rahm: Similarity Flooding: A Versatile Graph Matching Algorithm, In Proc 18th Int'l Conf. Data Eng. (ECDE'02) (2002) 117-128.
5. Julian Seidenberg and Alan Rector: Web Ontology Segmentation: Analysis, Classification and Use, WWW2006, (2006).
6. H. Stuckenschmidt and M. Klein: Structure-Based Partitioning of Large Class Hierarchies. In Proc of the 3rd International Semantic Web Conference (2004).
7. W. Hu, Y. Qu: Block matching for ontologies, In Proc of the 5th International Semantic Web Conference, LNCS, vol. 4273, Springer (2006) 300-313.
8. P.G. Xu, HJ. Tao: SOBOM: An Anchor Concept and Sub-ontology based Ontology Matching Approach. To be appear.

Cross-lingual Dutch to English alignment using EuroWordNet and Dutch Wikipedia

Gosse Bouma

Information Science, University of Groningen, g.bouma@rug.nl

Abstract. This paper describes a system for linking the thesaurus of the Netherlands Institute for Sound and Vision to English WordNet and dbpedia. We used EuroWordNet, a multilingual wordnet, and Dutch Wikipedia as intermediaries for the two alignments. EuroWordNet covers most of the subject terms in the thesaurus, but the organization of the cross-lingual links makes selection of the most appropriate English target term almost impossible. Using page titles, redirects, disambiguation pages, and anchor text harvested from Dutch Wikipedia gives reasonable performance on subject terms and geographical terms. Many person and organization names in the thesaurus could not be located in (Dutch or English) Wikipedia.

1 Presentation of the system

This paper describes our system for the very large cross-lingual resources (vlcr) task, which asked for an alignment between the thesaurus of the Netherlands Institute for Sound and Vision and English WordNet and (English) dbpedia, a database extracted from Wikipedia.

We used an ad-hoc system to achieve the alignment. For the mapping to English WordNet, we used EuroWordNet, a multilingual resource which contains a Dutch wordnet, as well as mappings from Dutch to English WordNet. For the mapping to dbpedia, we used page titles, redirects, and anchor texts harvested from Dutch Wikipedia, and mapped Dutch pages to English pages using cross-language links. Most XML preprocessing was done using XQuery. The alignment itself was done using (Sicstus) Prolog.

1.1 Background

For our work on open domain question answering, information extraction, and coreference resolution, we are interested in creating general, informal, taxonomies of entities encountered in Dutch texts.[1] As part of this work, we created a Dutch counterpart of the Yago system [4], in which Wikipedia categories are aligned with a Dutch wordnet [1]. We expected that the techniques we used there (especially stemming and parsing of labels, and using predominant word senses for sense disambiguation) could be applied to the present task as well.

[1] Some results can be found on www.let.rug.nl/gosse/Ontology

1.2 Aligning GTAA to WordNet via EuroWordNet

The mapping from the thesaurus of the Netherlands Institute for Sound and Vision (GTAA) and English Wordnet was accomplished using EuroWordNet [6]. We concentrated on the subset of the thesaurus that contained subject labels, as these are mostly common nouns or noun phrases headed by a common noun. The Dutch part of EuroWordNet (EWN) contains hardly any proper names, so we expected the overlap between EWN and the other parts of the thesaurus (on person names, geographical locations, and organizations) to be minimal.

The alignment procedure is schematically represented in figure 1.

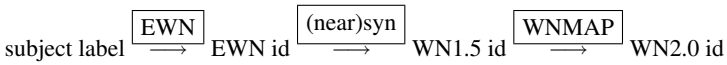


Fig. 1. Mapping GTAA to WordNet

Entries in the thesaurus are often plurals (a*fgevaardigden (representatives), spoorwegen (rail roads), autobussen (buses)*), whereas dictionary entries in EWN are typically singular. To ensure coverage of these cases, all entries in the subject part of the thesaurus were stemmed using the Alpino parser [5]. Alpino is a wide-coverage dependency parser for Dutch, which includes a morphological analyzer. As the analyzer also performs compound analysis (ie. *autobussen* is analyzed as *auto_bus*), we also parsed all EWN entries with Alpino. Thus, we can find a subject label in EWN by comparing stems. Note that compound analysis would also allow us to link a compound such as *bedrijfspionage (industrial espionage)* to a more general concept such as *espionage* (assuming a hypernym relation), but such links were not requested in the task definition.

EuroWordNet is a multilingual wordnet, in which each synset is linked to one or more inter language index ids (ILIs). ILIs in turn are linked to WordNet 1.5 ids. Links can express among others a synonym, near-synonym, hyponym or hypernym relation. We used only the synonym and near-synonym relations. Using the ILIs, each Dutch synset can be linked to an English WordNet id. As we will explain below, this step is in general one-to-many, as most Dutch synsets are connected to more than one ILI through the near-synonym relation. In the final step, we mapped WordNet 1.5 ids to WordNet 2.0 ids (the version of WordNet that was used to create the RDF/OWL version of WordNet that was the target of the mapping), using the WordNet mappings described in [2].[2]

1.3 Aligning GTAA to dbpedia via Dutch Wikipedia

For linking GTAA entries to dbpedia, we decided to use Dutch Wikipedia as intermediary, and to aim for linking GTAA entries to English Wikipedia pages. Translation of English Wikipedia pages into dbpedia URI's is done by means of a small script that adds the correct prefix, and deals with special characters.

[2] available from from `www.lsi.upc.es/~nlp/tools/mapping.html`

For our work on automatic annotation of web pages with links to Wikipedia [3], we had harvested a Dutch Wikipedia dump (august 2008) for cross-language links, redirects, disambiugation pages, and anchor texts (i.e. terms annoted on a Wikipedia page with a link to another Wikipedia page). We also used a list of English page names from a dump of English Wikipedia (january 2008).

The first step in the alignment is to generate all variants of a label. To match a term in the thesaurus with a Wikipedia page directly, for instance, it is necessary that the first letter is in upper case. Person names in GTAA are given as *Lastname, Firstname*, whereas Wikipedia simply uses *Firstname Lastname*. Subjects in GTAA are often plural, whereas they tend to be singular in Wikipedia. Singular forms are obtained from the parsed version of the subject labels that was also used in the aligment with WordNet. Finally, alternative labels provided by GTAA are considered as variants of the concept label.

For all variants of a GTAA concept label, we try to find a matching Dutch Wikipedia page. This can be achieved by an exact match with a Wikipedia page, by an exact match with a redirect page (in which case the target of the redirect is the desired Wikipedia page), by finding a matching anchor text (in which case the most frequent target page for that anchor is returned) or by an exact match with a disambiguation page (in which case all options are returned). Given a suitable Dutch page, we find the English page by following the cross-language link from Dutch to English Wikipedia. In some cases such a link was absent. If a Dutch page (with a corresponding English page) could not be found by means of the techniques above, we tried to find a matching page in English Wikipedia directly, using only page titles.

We expect that there will be a difference in accuracy between the various methods for finding an English page. Preference (and a high confidence score) is given to direct matches, followed by redirects, anchors, direct matches in English, and disambiguation pages.

1.4 Scripts and results

The scripts used to produce the alignment can be found at `www.let.rug.nl/gosse/GTAA`. Note that EuroWordNet data is missing, as this is a resource which is not in the public domain.

The results of our alignment can be found at `www.let.rug.nl/gosse/GTAA/bouma-vlcr.tgz`.

2 Results

2.1 vlcr: GTAA to WordNet

We only tried to link GTAA *subject* entries to WordNet. An overview of the results is given in table 1. Note that coverage is quite reasonable between GTAA and EWN. Where no link could be found, this is mostly due to multiword subject labels (such as *alternatieve energie (alternative energy)* or *bedreigde diersoorten (endangered species)*) and compounds. Multiword phrases are generally absent from EWN, and we made no

attempt to search for these in English WordNet directly. Other subjects that could not be linked often consist of a compound noun. As compounding is a productive process, we do not expect all compounds to be present in EWN. Given the fact that we do have a morphological analysis, we could have linked compound nouns to a more general concept (i.e. the head noun) by means of a hypernym link. Such links were not part of the task, however. Together, multiword phrases and compounds account for over 80% of the subject labels that could not be linked. 5% coverage was lost in the mapping from WordNet 1.5 to WordNet 2.0.

subject labels	3878
linked to EWN	2617 (67%)
unique ILIs	3703
avg. ambiguity	1.4
linked to WN2.0	2392 (62%)
unique synsets	3676
avg. ambiguity	1.5

Table 1. Alignment results for GTAA to EuroWordNet and WordNet 2.0

Ambiguity of the target is a serious problem. This is not only caused by the fact that a word may belong to more than one synset (word sense ambiguity), but also by the fact that the mapping between synsets in EWN and WN through ILI links is highly ambiguous. The Dutch nouns part of EWN contains only 631 synonym relation ILIs (which tend to be unique), and no less than 4641 near-synonym relation ILIs (which tend to link to several WN targets). One might consider reducing the ambiguity by selecting the most appropriate word sense for a given subject label. This is by no means trivial however (see [1] for some results for Dutch). In this particular case, it is also not very effective, as many synsets are themselves connected to more than one English synset through the near-synonym relation. The situation is illustrated in figure 2. The concept *brons* is linked to two synsets in EWN. As WN has two synsets for the *bronze* as well, one might expect each of these synsets to be linked to a specific WN synset. In reality, however, each EWN synset is linked to each WN synset. Thus, even if one resolved the concept *brons* to the correct EWN synset, it still would be practically impossible to decide which of the two WN synsets ought to be chosen (as the information on how to disambiguate synsets between wordnets is simply not given). In our results, both targets are given as possible alignment, but lower confidence is given to links involving a near-synonym relation.

2.2 vlcr: GTAA to dbpedia

Table 2 gives some results for linking the four different parts of the GTAA thesaurus (subject/concepts, names/organisations, locations, and persons) to English Wikipedia. Coverage is best for subjects and locations. GTAA contains many names of persons and

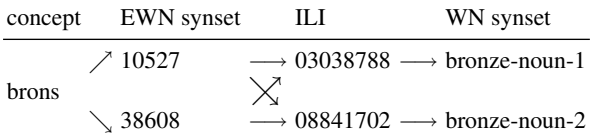


Fig. 2. Linking the concept *brons* to two EWN synsets, and two WN synsets.

organisations that seem to be absent in both Dutch and English Wikipedia. It should also be noted that coverage of location names is high only because many location names are found in English Wikipedia directly. This holds partly for names of organisations as well, but less so for person names. For 6 - 9% of the concepts, a Dutch Wikipedia target could be found, but no corresponding English page existed.

	subject		name		location		person	
link type	links	%	links	%	links	%	links	
nlpage	2027	52.3	3128	11.5	5135	36.7	7311	7.5
redirect	423	10.9	984	3.6	400	2.9	762	0.8
anchor	621	16.0	616	2.3	357	2.6	176	0.2
enpage	260	6.7	4085	15.1	3705	26.5	9246	9.5
linked	3127	**80.6**	8830	**32.6**	9602	**68.6**	17521	**17.9**
no-english	357	9.2	2197	8.1	878	6.3	5721	5.9
no-link	394	10.2	16077	59.3	3512	25.1	74375	76.2
total	3878		27104		13992		97617	

Table 2. Alignment results for GTAA to Dutch and English Wikipedia

3 Discussion

In general, it seems that even with relatively modest technology, a mapping between two resources in different languages can be achieved. It should be noted, however, that the mapping to WordNet owes much to the existence of EuroWordNet, which solves the most difficult (cross-language) part of the task to a large extent. On the other hand, EuroWordNet does not help much in deciding which synset for a given English term is the appropriate one.

Our results for Wikipedia linking could still be improved in a number of ways. We hardly employed categorical constraints. The GTAA thesaurus comes in four parts. Each part is a different category. This information could be used to block the link from *A4* in the locations file to *A4 (paper format)* in Wikipedia. Similarly, concept labels often come with a skope note. Word overlap could be used to select the correct target page (i.e. to prefer *highway A4 in the Netherlands* over that in *Austria*)). Alternatively,

one might use the information that concepts with the same scope note are likely to be linked to Wikipedia pages with identical or closely related Wikipedia categories to detect outliers. For selecting the most promising target, we experimented with a simple preference scheme (which always prefers the link given by the most reliable relation), and a simple weighting scheme (which adds scores when multiple links to the same target are found). Weighting was used for the final results. No doubt, more subtle schemes could be developed. For instance, at the moment we only take into account the most frequent target of an anchor text. Alternatively, one might consider all targets pointed to by anchor text as potential targets, and use the frequency of these links as a weight.

Somewhat surprisingly, we discovered that cross-language links are not reversible. Initially, we used cross-language links harvested from English Wikipedia, as this is the larger resource, and we expected that this might also be more thorough in providing cross-language links. However, since English Wikipedia has more pages than Dutch Wikipedia, several English pages may be linked to the same Dutch page (i.e. *Bowling* and *Ten pin Bowling* both point to the Dutch page *Bowling*). If one works with cross-language links harvested from Dutch Wikipedia, this situation does occur less frequently, although similar problems can occur here as well (i.e. in the versions of Wikipedia we used, the Dutch *A4 highway* was linked to an English page which redirected to a general page on Dutch highways).

4 Conclusion

We have presented a method for linking the thesaurus of the Netherlands Institute for Sound and Vision with two English resources, WordNet and Wikipedia. We used an ad-hoc method which relied on the existence of cross-language links for similar data, namely EuroWordNet, a multi-lingual wordnet with cross-language links, and Dutch Wikipedia, which contains cross-language links to English Wikipedia.

References

1. Gosse Bouma. Linking Dutch Wikipedia Categories to EuroWordNet. In *Proceedings of the 19th Computational Linguistics in the Netherlands meeting (CLIN 19)*. Groningen, the Netherlands, 2009.
2. J. Daude, L. Padro, and G. Rigau. Mapping wordnets using structural information. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 504–511. Association for Computational Linguistics Morristown, NJ, USA, 2000.
3. Proscovia Olango, Gerwin Kramer, and Gosse Bouma. Termpedia for interactive document enrichment. In *Computational Linguistics Applications (CLA) workshop at IMCSIT*, Mragowo, Poland, 2009.
4. Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 697–706, New York, NY, USA, 2007. ACM Press.
5. Gertjan van Noord. At last parsing is now operational. In Piet Mertens, Cedrick Fairon, Anne Dister, and Patrick Watrin, editors, *TALN06. Verbum Ex Machina. Actes de la 13e conference sur le traitement automatique des langues naturelles*, pages 20–42. 2006.
6. P. Vossen. Eurowordnet a multilingual database with lexical semantic networks, 1998.

TaxoMap in the OAEI 2009 alignment contest

Fayçal Hamdi[1], Brigitte Safar[1], Nobal B. Niraula[2], and Chantal Reynaud[1]

[1] LRI CNRS UMR 8623, Université Paris-Sud 11, Bat. G, INRIA Saclay
2-4 rue Jacques Monod, F-91893 Orsay, France
firstname.lastname@lri.fr
[2] nobal.niraula@inria.fr

Abstract. TaxoMap is an alignment tool which aims to discover rich correspondences between concepts. It performs an oriented alignment (from a source to a target ontology) and takes into account labels and sub-class descriptions. This new implementation of TaxoMap reduces significantly runtime and enables parameterization by specifying the ontology language and different thresholds used to extract different mapping relations. It improves terminological techniques, with a better use of TreeTagger and introduces new structural techniques which take into account the structure of ontology. Special effort has been made to handle large-scale ontologies by partitioning input ontologies into modules to align. We conclude the paper by pointing out the necessary improvements that need to be made.

1 Introduction

TaxoMap was designed to retrieve useful alignments for information integration between different sources. The alignment process is then **oriented** from ontologies that describe external resources (named *source* ontology) to the ontology (named *target* ontology) of a web portal. The target ontology is supposed to be well-structured whereas source ontology can be a flat list of concepts.

TaxoMap makes the assumption that most semantic resources are based essentially on classification structures. This assumption is confirmed by large scale ontologies which contain rich lexical information and hierarchical specification without describing specific properties or instances.

To find mappings in this context, we can only use the following available elements: labels of concepts and hierarchical structures.

The new implementation of TaxoMap proposes a better morpho-syntactic analysis and new techniques. Moreover, the methods to partition large ontologies into modules which TaxoMap can handle easily were refined.

We take part to five tests. We hope we perform better in terms of precision of mappings generated and runtime. Tests on library data sets allow us to experiment our algorithm on large multilingual ontologies (English, French, and German).

2 Presentation of the System

2.1 State, Purpose and General Statement

We consider an ontology as a pair (C, H_C) consisting of a set of concepts C arranged in a subsumption hierarchy H_C. A concept c is defined by two elements: a set of labels and subclass relationships. The labels are terms that describe entities in natural language and which can be an expression composed of several words. A subclass relationship establishes links with other concepts.

Our alignment process is oriented; from a source (O_S) to a target (O_T) ontology. It aims at finding one-to-many mappings between single concepts and establishing three types of relationships, equivalence, subclass and semantically related relationships defined as follows.

Equivalence relationships An equivalence relationship, *isEq*, is a link between a concept in O_S and a concept in O_T with labels assumed to be similar.

Subclass relationships Subclass relationships are usual *isA* class links. When a concept c_S of O_S is linked to a concept c_T of O_T with such a relationship, c_T is considered as a super concept of c_S.

Semantically related relationships A semantically related relationship, *isClose*, is a link between concepts that are considered as related but without a specific typing of the relation.

2.2 Techniques Used

The different techniques are based on the use of the moropho-syntactic analysis tool TreeTagger [1], and a similarity measure which compares the trigrams of the concept labels [2].
TreeTagger is a tool for tagging text with part-of-speech and lemma information, enables to take into account the language, lemma and an use word categories in an efficient way. The words are classified as functional (verbs, adverbs or adjectives) and stop words (articles, pronouns). Once classified by TreeTagger, the words are divided into two classes, **full words** and **complementary words**, according to their category and their position in the label. In principle, all names are full words except if they are placed after a determiner, all other words are complementary words.
This classification is then used to give more weight to the full words in the calculation of similarity between labels.

The main methods used to extract mappings between a concept c_s in O_S and a concept c_t in O_T are:

– Label equivalence: An equivalence relationship, *isEq*, is generated if the similarity between one label of c_s and one label of c_t is greater than a threshold (Equiv.threshold).

- High lexical similarity: Let c_{tmax} be the concept in O_T with the highest similarity measure with c_s. If the similarity measure is greater than a threshold (HighSim.threshold) and if one of the labels of c_{tmax} shares at least two full words in common with one of the labels of c_s, the heuristic generates the relationship $< c_s\ isA\ c_{tMax} >$ if the label of c_{tmax} is included in the c_s one, otherwise it generates $< c_s\ isClose\ c_{tMax} >$.
- Label inclusion (and its inverse): If one of the labels of c_{tmax} is included in one of the labels of c_s, and if all words of included label are full words, we propose a subclass relationships $< c_s\ isA\ c_{tmax} >$. Inversely, if one of the labels of c_s is included in one of the labels of c_{tmax}, we propose a semantically related relationships $< c_s\ isClose\ c_{tmax} >$.
- Reasoning on similarity values: Let c_{tMax} and c_{t2} be the two concepts in O_T with the highest similarity measure with c_s, the relative similarity is the ratio of c_{t2} similarity on similarity c_{tMax}. If the relative similarity is lower than a threshold (isA.threshold), one of the three following techniques can be used:
  - the relationship $< c_s\ isClose\ c_{tMax} >$ is generated if one of the labels of c_s is included in one of the labels of c_{tMax}, and the words of the included label are complementary words.
  - the relationship $< c_s\ isClose\ c_{tMax} >$ is generated if the similarity of c_{tMax} is greater than a threshold (isClose.thresholdMax).
  - an isA relationship is generated between c_s and the father of c_{tMax} if the similarity of c_{tMax} is greater than a second threshold (isA.thresholdMax).
- Reasoning on structure:
  - an isA relationship $< c_s\ isA\ c_t >$ is generated if the subclass relation $< c_s\ isSubClassOf\ X >$ appears in O_S and if the equivalence mapping $< X\ isEq\ c_t >$ have been identified.
  - the relationship $< c_s\ isClose\ c_t >$ is generated if c_t is the concept in O_T which have the most number of children in O_T with the same label as the children of c_s in O_S. More details of this approach are given at the end of this sub-section.
  - an isA relationship $< c_s\ isA\ p >$ is generated if the three concepts in O_T with the highest similarity measure with c_s have similarity greater than a threshold (Struct.threshold), and have a common father p in O_T.

As we mentioned above, we use a structural heuristic based on the $Semantic\ Cotopy$ measure of a concept, proposed by Maedche and Staab [3]. The $Semantic\ Cotopy$ is based on the intentional semantics of a concept c in an ontology O, $SC(c, O)$, defined as the set of all its super- and sub-concepts in O. When a concept c belongs to two ontologies, one can define the taxonomic overlap (TO) between O_1 and O_2 for this concept, denoted $TO(C, O_1, O_2)$ and defined as the ratio between the number of common elements in the intentional semantics of c in O_1 and in O_2 and the total number of elements belonging to the union of these two sets. If a concept c is in O_1 but not in O_2, an optimistic approximation of $TO(c, O_1, O_2)$ is defined as the maximum overlap obtained by comparing $SC(c, O_1)$ to the intentional semantics of all the concepts in O_2. Our heuristic uses $SC_D(c)$ which includes only the concept and its descendants instead of the original Semantic Cotopy. If a concept c is in O_1 but not in O_2, we propose as candidate mapping for this concept c, the concept c_{Max} of O_2 which maximizes the TO, if c and c_{Max} have at least two descendants in common.

2.3 Partitioning of large scale ontologies

We propose a method of ontology partitioning [4], that relies on the implementation of PBM [5] algorithm. PBM partitions large ontologies into small blocks (or modules) and constructs mappings between the blocks, using predefined matched class pairs, called *anchors* to identify related blocks. We reuse the partitioning part and the idea of anchors, but the originality of our method, called PAP (*Partition, Anchor, Partition*), is that it is *alignment oriented*, that means that the partitioning process is influenced by the mapping process.

The PAP method consists of:

- decompose the most structured ontology, that will be called the *target*, O_T, into several blocks B_{Ti}, according to the PBM algorithm.
- force the partitioning of the other ontology, called the *source* O_S, to follow the pattern of O_T. To achieve this, the method identifies for each block B_{Ti} constructed from O_T all the anchors belonging to it. Each of these sets of anchors will constitute the kernel or *center* CB_{Si} of a future block B_{Si} which will be generated from the source O_S.
- reuse the PBM algorithm to partition the source O_S around the centers CB_{Si}.
- align each block B_{Si} built from a center CB_{Si} with the corresponding block B_{Ti}.

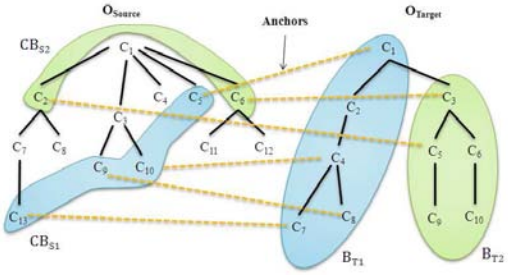


Fig. 1. *The centers CB_{Si} identified from B_{Ti}*

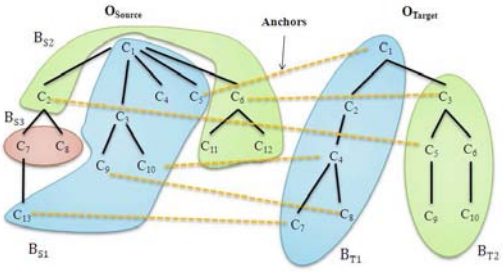


Fig. 2. Partition of O_S around the centers CB_{Si}

The tests show that the maximum size of the blocks has to be fixed for the target ontology. If the themes covered by both ontologies are of the same importance, i.e. if the source ontology corresponds to a representation of the same importance than the representation of the target one, a maximum size for the blocks in the source ontology is not needed. Their size will become close to the size of the blocks of the target ontology. This phenomenon allows to avoid obtaining a lot of small isolated blocks which appear when the maximum size of the blocks of the source ontology is fixed.

So, on the example of Fig2, the B_{S3} block remains isolated because the size of of the source blocks was fixed. Without limitation of the size, the B_{S3} block can be merged with B_{S2}. The only blocks which will remain isolated will be the blocks built

when the source ontology will be partitioned, independently of the kernels identified in the decomposition of the target ontology, i.e. concepts with no relation with those of the target ontology. So, the fact that the concepts belonging to these isolated blocks are not aligned should not damage our results.

2.4 Adaptations made for the Evaluation

Unlike in previous years, we have made some specific adaptations for the OAEI 2009 campaign.

For Anatomy task, we did not use the techniques which generate isA relationship. All the alignments outputted by TaxoMap are uniformly based on the same parameters. We had, however, fixed confidence values depending on relation types.

For library test, data sets consist of multilingual ontologies. In order to use lexical comparison, we translated non-English labels of all of the concepts of the vocabularies into English. The translation is done by using Googles translation APIs.

2.5 Link to the system and parameters file

TaxoMap requires:

- Mysql [3]
- Java (Version 1.5 and above)[4]
- Google's Java Client API for Translation [5]
- TreeTagger with its language parameter files [6]

The version of TaxoMap (with parameter files) used in 2009 contest can be downloaded from:

- http://www.lri.fr/~hamdi/TaxoMap.jar: a parameter lg has to be specified it denotes the language of the ontology. For example TaxoMap.jar *fr* to perform alignment on ontologies in French. If no language is specified, it is supposed to be English.
- http://www.lri.fr/~hamdi/TaxoMap.properties: a parameter file which specifies:
  - The command to launch TreeTagger.
  - TreeTagger word categories that has to be considered as functional, stop words and prepositions.
  - The RDF output file.
  - Different thresholds of similarity, depending on the method used.
- http://www.lri.fr/~hamdi/dbproperties.properties: a parameter file which contains the user and password to access to MySql.

[3] http://www.mysql.com
[4] http://java.sun.com
[5] http://code.google.com/p/google-api-translate-java
[6] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger

2.6 Link to the Set of Provided Alignments

The alignments produced by TaxoMap are available at the following URLs:
http://www.lri.fr/~hamdi/benchmarks/
http://www.lri.fr/~hamdi/anatomy/
http://www.lri.fr/~hamdi/directory/
http://www.lri.fr/~hamdi/library/
http://www.lri.fr/~hamdi/benchmark-subs/

3 Results

3.1 Benchmark Tests

Since our algorithm only considers labels and hierarchical relations and only provides mapping for concepts, the recall would have been low even for the reference alignment. The overall results would have been similar -with no surprise- to those of last year.

3.2 Anatomy Test

The anatomy real world case is to match the Adult Mouse Anatomy (denoted by *Mouse*) and the NCI Thesaurus describing the human anatomy (tagged as *Human*). *Mouse* has 2,744 classes, while *Human* has 3,304 classes. As last year, we considered *Human* as the target ontology as is it well structured and larger than *Mouse*.

TaxoMap performs the alignment (with no need to partition) in about 8 minutes which is better than last year [6] where TaxoMap took about 25 minutes to align the two ontologies.

As only equivalence relationships will be evaluated in the alignment contest, we did not use this year the techniques which generate isA relationship (except in the Task 3) and we change *isClose* mapping to equivalence. As a result, we found fewer mappings than last year but we hope that the precision will be better.

- For the first task, TaxoMap discovers 1274 mappings, 973 Equivalence relations and 301 Proximity relations.
- For the second task, we got only 1084 mappings, 973 Equivalence relations and 111 Proximity relations, using only the heuristic which identifies the relation $< c_s$ $isClose$ $c_{tMax} >$ when one of the labels of c_s is included in one of the labels of c_{tMax}.
- For the third task, we used, in addition of the techniques ot the first task, the heuristic which identifies subsumption links with "High Lexical Similarity". This allows to discover 1451 mappings and to slightly increase the recall, but reduce the precision. In fact, many mappings like <hand blood vessel isA Blood Vessel > or <iris blood vessel isA Blood Vessel > are semantically correct but become false when the subsumption relation isA is automatically replaced by an Equivalence relation.

– For the fourth task, we used the partial reference mapping in our partitioning method and we obtained 1131 mappings. This lower number of mapping is explained by two facts. The first one is that the structural heuristic based on the $Semantic\ Cotopy$ is the only one of which the results can be improved by the use of the partial mapping. The second one is that the partitioning method increases the precision but reduces the recall.

3.3 Directory Test

The directory task consists of Web sites directories like Google, Yahoo! or Looksmart. To date, it includes 4,639 tests represented by pairs of OWL ontologies. TaxoMap takes about 40 minutes to complete all the tests.

3.4 Library Test

In order to use lexical comparison in library data sets, which consist of multilingual ontologies, we used Google translation API [7] to translate non-English labels into English. With our current configuration, we cannot partition the large sized library ontologies. However, we used just a part of its data set to partition and then to find the mappings among concepts.

As skos relations will be evaluated, we change different mapping types to skos ones with these confidence values:

– (type1) *isEq* relations become skos:exactMatch with a confidence value set to 1.
– (type2) *isA* relations become skos:narrowMatch with a confidence value set to 1 for label inclusion, 0.5 for relations generated by structural technique or by relative similarity method.
– (type3) *isGeneral* relations become skos:broadMatch with a confidence value set to 1.
– (type4) *isClose* relations become skos:relatedMatch with a confidence value set to 1.

Generated mappings are as follows:

– ***LCSH-RAMEAU:*** 5074 *type1* relations, 48817 *type2* relations, 116789 *type3* relations and 13205 *type4* relations.
– ***RAMEAU-SWD:*** 1265 *type1* relations, 6690 *type2* relations, 17220 *type3* relations and 1317 *type4* relations.
– ***LCSH-SWD:*** 38 *type1* relations.

3.5 Benchmark-Subs Test

Benchmark-Subs tests aims to evaluate alignments which contain other mapping relations than equivalence. Two tasks are available in this test: Gold-standard based evaluation concerning the evaluation of subsumption relations and open-ended task concerning the evaluation of equivalence and non-equivalence mappings. In our tool, for the first task, we use lexical methods to obtain subsumption relations.

4 General Comments

4.1 Results

The new version of TaxoMap improves significantly the results on the previous version of TaxoMap in terms of runtime and precision of generated mappings. The new implementation offers extensibility and modularity of code. TaxoMap can be parameterized by the language used in ontologies, the choice of used techniques and different thresholds. Our partitioning algorithms allow us to participate to tests with large ontologies.

4.2 Future Improvements

The following improvements can be made to obtain better results:

- To take into account all concepts properties instead of only the hierarchicals ones.
- Use of WordNet as a dictionary of synonymy. The synsets can enrich the terminological alignment process if an *a priori* disambiguation is made.
- To develop the remaining structural techniques which proved to be efficient in last experiments [8] [9].

5 Conclusion

This paper reports our participation to OAEI campaign with the new implementation of TaxoMap. Our algorithm proposes an oriented mapping between concepts. Due to partitioning, it is able to perform alignment on real-world ontologies. Our participation in the campaign allows us to test the robustness of TaxoMap, our partitioning algorithms and new structural techniques.

References

[1] Schmid H. Probabilistic Part-of-Speech Tagging Using Decision Trees, International Conference on New Methods in Language Processing (1994)

[2] Lin, D. : An Information-Theoretic Definition of Similarity. ICML. Madison. (1998) 296–304

[3] Maedche, A. and Staab S. Measuring Similarity between Ontologies, EKAW (2002)

[4] Hamdi, F., Safar, B., Reynaud, C. and Zargayouna, H. Alignment-based Partitioning of Large-scale Ontologies, in Advances in Knowledge Discovery and Management (AKDM09), to appear.

[5] Hu, W., Zhao, Y., and Qu, Y. Partition-based block matching of large class hierarchies, Proc. of the 1st Asian Semantic Web Conference (ASWC06). pp.72-83, (2006)

[6] Hamdi, F., Zargayouna, H., Safar, B., and Reynaud, C. TaxoMap in the OAEI 2008 alignment contest, Proceedings of the ISWC'08 Workshop on Ontology Matching OM-08 (2008)

[7] http://code.google.com/p/google-api-translate-java/

[8] Reynaud, C. and Safar, B. When usual structural alignment techniques don't apply, The ISWC'06 Workshop on Ontology matching (OM-06), (2006)

[9] Reynaud, C. and Safar, B. Exploiting WordNet as Background Knowledge,The ISWC'07 Workshop on Ontology Matching (OM-07), (2007)

Using Ontology Alignment to Dynamically Chain Web Services

Dru McCandless,* Leo Obrst
{mccandless, lobrst}@mitre.org
The MITRE Corporation

Abstract. This statement of interest presents a brief rationale and description of issues for using ontology alignment as a key step in dynamically chaining together a sequence of web services.

Keywords: Ontology Alignment, Web Services, Dynamic Service Chain Composition

1 Introduction

As in much of the world, the Department of Defense (DoD) has seen an explosion in the growth of web services. But integration of these disparate information sources to answer complex questions remains a challenge. Many information integration tasks are unforeseen at the time the services are constructed, and are therefore difficult to perform "on the fly". This typically involves searches among various web service definitions and deciding how best to arrange and call them in an ad-hoc manner. A better method of assembling a dynamic service chain is needed.

Using semantic web technology to semi-automatically create a service chain is an active area of research [2, 3, 4, 5]. However, most of this work is centered on the use of formal ontologies using standards such as WSMO (Web Services Modeling Ontology), WSML (Web Services Modeling Language), OWL-S, or SAWSDL (Semantic Annotations for WSDL), which assume that the builders of web services will also build the accompanying ontologies necessary for integration. This hasn't been the case for DoD web service builders. This is in part because there is a lack of consensus about ontology standards, and the skills needed to develop ontologies are different from those needed to build and deploy web services. As a result, these services do not have formal ontologies that define the domain within which the service operates or that describe the service messages.

We have developed a different approach for dynamic web service assembly that takes advantage of the formal structure inherent in web services that are defined by WSDL documents. This is based on our past efforts using ontology alignment to integrate different sources of information [1, 6]. The XML Schema definitions are extracted from the WSDLs, and the schemas are then converted into OWL. The resulting OWL files are aligned using ontology alignment tools, which allows for

* Communicating author: Dru McCandless, The MITRE Corporation, 1155 Academy Park Loop, Colorado Springs, Colorado 80910.

semi-automated mapping of the service input and output messages at the semantic level. A theorem prover is then used to construct a service chain based on the aligned service inputs and outputs which meets some information goal.

2 Issues

We are sometimes asked to justify converting schemas into ontologies to do alignment when there are schema alignment tools available. Our response is that by converting to an ontology, it enables us to apply the power of the underlying logic model to make better decisions – an example being the case where there are two schemas with the word 'mustang', but one refers to the car and the other the horse. A purely linguistic aligner will almost always align these – and usually the schemas we work with are small enough such that a structural analysis doesn't have enough information to make a better decision. But by using ontologies, it is fairly easy for a person to add some additional taxonomy information above each of the 'mustang' classes, by asserting for example that one mustang is a subclass of vehicle and the other is a subclass of animal, and that the two classes are disjoint. This should enable an ontology matcher to reach the correct conclusion. In addition, of course, using an ontology enables one to perform automated consistency checking on it – something that is not easy to do with a schema. As alignments become complicated, and the ontologies involved become large and complex, consistency checking becomes increasingly valuable.

In practice, when applying ontology alignment to real-world services a number of difficulties are encountered, with missing and false alignments being the most frequent. Some of the areas about which we would like to engage alignment researchers are: an increased emphasis on meta-properties, such as disjointness, to help with alignment decisions, and techniques for including domain information (such as mid-level ontologies or controlled vocabularies) to improve performance.

References

1. Stoutenburg, S., Hawthorne S., Obrst L., McCandless D., Wootton S., Bankston B., Rapid, Agile Integration Through Ontology Mapping. MITRE 2008 Technical Report, forthcoming.
2. Hibner A., Zielinski K., "Semantic-based Dynamic Service Composition and Adaptation," pp.213-220, 2007 IEEE Congress on Services (Services 2007), 2007
3. Agre, Marinova, "An INFRAWEBS Approach to Dynamic Composition of Semantic Web Services," Bulgarian Academy of Sciences, Cybernetics and Information Technologies Vol 7, Number 1.
4. Traverso P., Pistore M., Automated composition of semantic web services into executable processes. In 3rd International Semantic Web Conference (ISWC 2004), pages 380–394, 2004.
5. Rao J., Su X., "A Survey of Automated Web Service Composition Methods". In Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, SWSWPC 2004, San Diego, California, USA, July 6th, 2004.
6. McCandless, Dru; Leo Obrst; Shayn Hawthorne. 2009. Dynamic Web Service Chaining using OWL and a Theorem Prover. Accepted, Third IEEE International Conference on Semantic Computing, Berkeley, CA, USA - September 14-16, 2009.

Semantic geo-catalog: a scenario and requirements

Pavel Shvaiko[1], Lorenzino Vaccari[2], and Gaia Trecarichi[3]

[1] TasLab, Informatica Trentina S.p.A., Trento, Italy
[2] Autonomous Province of Trento, Trento, Italy
[3] DISI, University of Trento, Trento, Italy

Abstract. In this short paper we present a scenario and requirements for ontology matching posed by a geographical application, namely a semantic geo-catalog, which is an integral part of any spatial data infrastructure (SDI). It enables semantic interoperability among various geo-data and geo-service providers, and thus, contributes to the harmonization of geo-information.

Introduction. The need for coherent and contextual use of geographic information between different stakeholders, such as departments in public administrations, formed the basis for a number of initiatives aiming at sharing of spatial information, e.g., the INfrastructure for SPatial InfoRmation in Europe (INSPIRE)[1], see also [8, 10]. In this paper, we focus on a particular component of the INSPIRE architecture, which is a discovery service, that ought to be implemented by means of the Catalogue Service for the Web (CSW)[2], a recommendation of the Open Geospatial Consortium (OGC).

There have been provided several implementations of the CSW-based geo-catalog, e.g., GeoNetwork[3]. A first attempt to provide a semantic geo-catalog has been made in [6], though it was based on a single ontology approach. The approach in [9] proposed an OWL profile for CSW. Finally, the 52°North semantics community[4] proposed to encapsulate ontology repositories by OGC services. In turn, the problem of ontology matching [2] in geo applications has been rarely addressed [7], with some exceptions, such as in [1, 5, 10]. The contribution of this paper includes a specific scenario and requirements for ontology matching posed by a semantic geo-catalog to be realized within the SDI of the Autonomous Province of Trento (PAT).

Scenario. Figure 1 shows a high-level architecture for the semantic geo-catalog system-to-be. Users can issue queries, such as *Trentino mountain hovels reachable with main roads*. The query and search results, such as a map of hovels, are handled by the Trentino geo-portal[5] implemented within the BEA ALUI framework. The geo-catalog will be based on the GeoNetwork open source, while its semantic extension will be designed and developed on top of SWeb[6] search and matching technologies [3, 4]. Specifically, user queries will be analyzed in order to extract concepts out of labels. Then, these are matched at run time against the *universal knowledge* of the SWeb system (SWebUK). In turn, GeoNetwork will contain domain specific ontologies (e.g., Agrovoc) which are associated with geo-metadata and matched with the SWebUK at design time.

[1] http://www.ec-gis.org/inspire/
[2] http://www.opengeospatial.org/standards/cat
[3] http://geonetwork-opensource.org/
[4] http://52north.org
[5] http://www.territorio.provincia.tn.it/
[6] http://www.dit.unitn.it/~knowdive/description.php

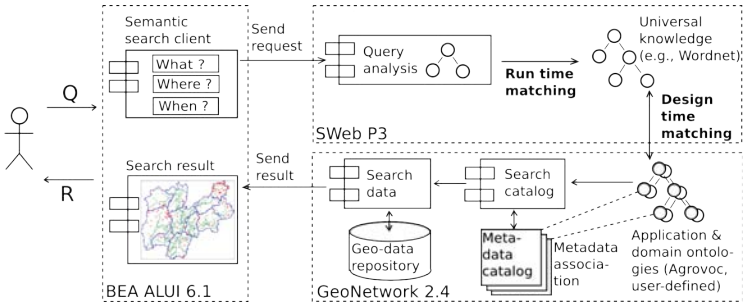


Fig. 1: A high-level architecture for geo-service discovery.

Requirements. There are six general key requirements indicated by INSPIRE, three of which are going to be monitored (for the discovery service), such as: *performance* - to send one metadata record within 3s.; *availability* - service up by 99% of time; *capacity* - 30 simultaneous service requests within 1s. These requirements only put constraints on run time matching needed between the user query and ontologies of the system, that is, the time elapsed between query issue and search results (metadata records) returned should be at most of 3s., etc. Matching results can be approximate here, though their correctness (precision) is preferred over completeness (recall). As for the design time matching between the SWebUK and the domain specific ontologies of the geo-catalog, it can be performed off-line (semi-automatically with sound and complete alignment) when any of these knowledge sources evolves.

Conclusions and future work. In this short paper we have presented a scenario and requirements for ontology matching within a geo-information application, which is a semantic geo-catalog. Future work proceeds at least in the following directions: (i) formalization and in-depth study of the scenario, and (ii) implementation and evaluation of the semantic geo-catalog in order to bring it to production in the SDI of PAT.

Acknowledgments. This work has been supported by the *TasLab network* project funded by the European Social Fund under the act n. 1637 (30.06.2008) of PAT.

References

1. I. Cruz and W. Sunna. Structural alignment methods with applications to geospatial ontologies. *Transactions in Geographic Information Science*, 12(6):683–711, 2008.
2. J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, 2007.
3. F. Giunchiglia, U. Kharkevich, and I. Zaihrayeu. Concept search. In *Proc. of ESWC*, 2009.
4. F. Giunchiglia, F. McNeill, M. Yatskevich, J. Pane, P. Besana, and P. Shvaiko. Approximate structure-preserving semantic matching. In *Proc. of ODBASE*, 2008.
5. K. Janowicz, M. Wilkes, and M. Lutz. Similarity-based information retrieval and its role within spatial data infrastructures. In *Proc. of GIScience*, 2008.
6. P. Maué. An extensible semantic catalogue for geospatial web services. *Journal of Spatial Data Infrastructures Research*, 3:168–191, 2008.
7. P. Shvaiko and J. Euzenat. Ten challenges for ontology matching. In *Proc. of ODBASE*, 2008.
8. P. Smits and A. Friis-Christensen. Resource discovery in a European Spatial Data Infrastructure. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):85–95, 2007.
9. K. Stock, M. Small, Y. Ou, and F. Reitsma. OGC catalogue services - OWL application profile of CSW. Technical report, Open Geospatial Consortium, 2009.
10. L. Vaccari, P. Shvaiko, and M. Marchese. A geo-service semantic integration in spatial data infrastructures. *Journal of Spatial Data Infrastructures Research*, 4:24–51, 2009.

Tax and Revenue Service scenario for Ontology Matching

Stefano Brida[1], Marco Combetto, Silvano Frasson[2] and Paolo Giorgini[3]

[1] Trentino Riscossioni S.p.A., Trento, Italy
[2] Informatica Trentina S.p.A., Trento, Italy
[3] D.I.S.I., University of Trento, Italy

Abstract. In this paper we present a scenario for ontology matching posed by the Trentino Riscossioni S.p.A data integration system focusing the opportunity to enhance the level of data integration over a large set of Tax and Revenue industry-specific data sources.

Introduction. The mission of Trentino Riscossioni S.p.A[1], a company owned by the Autonomous Province of Trento, is to promote simplification processes and harmonize the activity of more than 250 public entities in the province, creating policies for fair taxation and for operating costs reduction. The need for consistent and contextual use of the heterogeneous information sources between its offices, the municipalities and the other public bodies is a fundamental requirement for the implementation of an accurate and balanced taxation system. In this paper we want to focus on the possibility offered by matching technology [1] to enhance the in the present day data integration architecture and increase its flexibility in managing hundreds of new data sources with reduced software development for each new sources added. Besides, even if the data integration has been extensively studied in the database community, according to some recent research works [2,3,4,5], the issue to improve the automatic schema matching in a data integration scenario for the Tax and Revenue market is a relative new ground of application. The contribution of this paper includes a specific scenario focusing several of the basic requirements that have to be considered in order to build a data integration system capable to support dynamically hundreds of data sources.

Scenario. The scenario is to make possible the insertion, management and deletion of new data sources (e.g., new data source from a new provincial database). The inclusion of a new data source would result in the census of syntactic and semantic information related to the attributes of the source and in an automatic mapping of these attributes over the proper attributes of the destination database schema. If the attributes are not present in the destination schema, the system must support the design of a schema extension. The source information is collected in a knowledge base. The search results will be available for at least 2 types of applications: (i) the business intelligence application that enables the monitoring, tracking and management of the data quality [6] of the integrated database and four (ii) mission-critical applications focusing specific business-strategic tasks: assessment revenue, territory mapping, planning support, final users services. As depicted hereafter in Figure 1, the information coming from the external data sources is processed through the SSMB (Semantic Schema/data Matching Box). The SSMB must be able to calculate the new system status $n + 1$ through a function based on the previous states

[1] http://www.trentinoriscossionispa.it

(*n*, *n-1*) in order to support a GUI tool that will provide the interface to the required information to the Information Engineer and to the calculated matching suggestions enabling to integrate the sources more rapidly than currently. There are about 10 different data sources for each municipality and 7-8 for each provincial data source. In the next 2 years, the plan is to integrate about 200 municipalities and other significant sources.

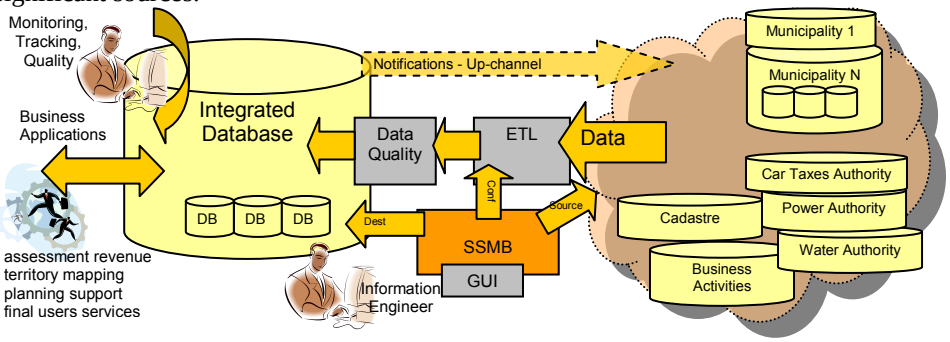


Figure 1 – The scenario description

The process analysis and breakdown provides confidence to motivate an implementation based on the use of a schema matching workbench like the HARMONY[7] integration workbench. In fact, beside the other advantages this approach enables the interoperation and the selection between different and various prototypes and commercial tools for schema matching and enables the sharing a common knowledge repository.

Conclusions and future works. We presented the business scenario for a solution that leverages on matching technology in order to scale-out over hundreds of data sources. Future works proceed in the following directions: (i) formalization of the scenario, (ii) evaluation and test of the HARMONY workbench features, and (iii) development of a specific working prototype for Trentino Riscossioni S.p.A.

Acknowledgments. This work has been supported by Trentino Riscossioni S.p.A and by Informatica Trentina S.p.A.-TasLab Network Project funded by the EU FSE under the act n. 1637 (30.06.2008) of the Autonomous Province of Trento.

References

[1] J. Euzenat and P. Shvaiko. Ontology matching. Springer, 2007.
[2] A. Bitterer, M. A. Beyer, and T. Friedman. Magic Quadrant for Data Integration Tools. Gartner, 2008.
[3] P. Shvaiko and J. Euzenat. Ten Challenges for Ontology Matching. In Proc. of ODBASE, 2008.
[4] K. Smith, P. Mork, L. Seligman, A. Rosenthal, M. Morse, C. Wolf, D. Allen, M. Li: The Role of Schema Matching in Large Enterprises. In Proc. of CIDR, 2009.
[5] Y. Asnar, P. Giorgini, P. Ciancarini, R. Moretti, M. Sebastianis, N. Zannone. Evaluation of Business Solutions in Manufacturing Enterprises. In International Journal on Business Intelligence and Data Mining, Inderscience, 2008
[6] Jeffery G. Watson. Data Quality Essentials. Uni. of Wisconsin-Madison, 2006
[7] P. Mork, L. Seligman, A. Rosenthal, J. Korb, and C. Wolf. The Harmony Integration Workbench. Journal on Data Semantics, 2008.

An Ontology-based Data Matching Framework: Use Case Competency-based HRM

Peter De Baer[1], Yan Tang[1], Pieter De Leenheer[2]
[1] VUB STARLab, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
[2] Collibra nv/sa, Ransbeekstraat 230, 12 Brussels, Belgium
{Peter.De.Baer, Yan.Tang}@vub.ac.be; Pieter@Collibra.com

Abstract. As part of the European PROLIX (Process Oriented Learning and Information eXchange) project, VUB STARLab designed a generic ontology-based data matching framework (ODMF). Within the project, the ODMF is used to calculate the similarity between data elements, e.g. competency, function, person, task, and qualification, based on competency-information. Several ontology-based data matching strategies were implemented and evaluated as part of the ODMF. In this article we describe the ODMF and discuss the implemented matching strategies.

Keywords: data matching, competency management, matchmaking, ontology

1 ODMF

Semantic data matching plays an important role in many modern ICT systems. Examples are data mining [6], electronic markets [1], HRM [2], service discovery [5], etc. Many existing solutions, for example [2], make use of description logics and are often tightly linked to certain ontology engineering platforms and/or domains of data matching. This often leads to a knowledge bottleneck because many potential domain users and domain experts may not be familiar with description logics or the specific platform at hand. To avoid such potential technical barrier we designed the ODMF so that it is independent of a particular ontology engineering platform, and does not require the use of description logics. Instead, we make use of the combination of an ontologically structured terminological database [3] and a DOGMA ontology [4] to describe data. Both the DOGMA ontology and the terminological database make use of natural language to describe meaning. On top of this semantic data model we developed an interpreter module and a comparison module. Both the interpreter and the comparator make use of a library of matching algorithms. The matching algorithms have access to the data model via an API, and may be written in any programming language that can access this Java API. Via the terminology base, data can be described and interpreted in different natural languages. We believe that this multilingualism will improve the usefulness of the framework within an international setting.

The ODMF is designed to support data matching in general. Currently, the ODMF has been, however, only implemented and evaluated as part of the European

integrated PROLIX project[1]. Within the PROLIX platform[2], the ODMF supports semantic matching of competency-based data elements, e.g. competency, function, person, task, and qualification.

2 Matching strategies

We implemented and evaluated several ontology-based data matching algorithms within the ODMF. These algorithms relate to three major groups: (1) string matching, (2) lexical matching, and (3) graph matching. However, most matching algorithms make use of a combination of these techniques.

1. *String matching techniques* are useful to identify data objects, e.g. competences and qualifications, using a (partial) lexical representation of the object. We selected two matching tools for this type of data matching: (a) regular expressions and (b) the SecondString[3] library.
2. *Lexical matching techniques* are useful to identify data objects, e.g. competences and qualifications, using a (partial) lexical representation of the object. In addition to plain string matching techniques, linguistic information is used to improve the matching. We selected two techniques to improve the matching: (a) tokenization and lemmatization and (b) the use of an ontologically structured terminological database.
3. *Graph matching techniques* are useful (a) to calculate the similarity between two given objects and (b) to find related objects for a given object.

References

1. Agarwal, S., Lamparter, S.: SMART - A Semantic Matchmaking Portal for Electronic Markets. In: Proceedings of the Seventh IEEE International Conference on E-Commerce Technology, pp. 405 – 408, (2005)
2. Biesalski, E., Breiter, M., Abecker, A.: Towards Integrated, Intelligent Human Resource Management. In: 1st workshop "FOMI 2005", Formal Ontologies Meet Industry (2005)
3. De Baer, P., Kerremans, K., and Temmerman, R.: Constructing Ontology-underpinned Terminological Resources. A Categorisation Framework API. Proceedings of the 8th International Conference on Terminology and Knowledge Engineering, Copenhagen (2008)
4. Jarrar, M., Meersman, R.: Formal Ontology Engineering in the DOGMA Approach. In: On the Move to Meaningful Internet Systems: CoopIS, DOA, and ODBASE, LNCS, Springer Verlag, pp. 1238-1254 (2002)
5. Shu, G., Rana, O. F., Avis, N. J., Dingfang, C.: Ontology-based semantic matchmaking approach. In: Advances in Engineering Software 38, pp. 59–67, ScienceDirect (2007)
6. Stamou, S., Ntoulas, A., and Christodoulakis, D.: TODE- an ontology based model for the dynamic population of web directories. In: "Data Mining with Ontologies: Implementations, Findings and Frameworks", edited by Nigro, H., O., Cisaro, S., E., G., Xodo, D., H., IGI Global (2007)

[1] http://www.prolixproject.org/
[2] http://prolixportal.prolix-dev.de/
[3] http://secondstring.sourceforge.net/

Improving bio-ontologies matching using types and adaptive weights

Bastien Rance[1] and Christine Froidevaux[1]

LRI, UMR 8623, Univ. of Paris-Sud, CNRS
F-91405 Orsay CEDEX France
firstname.surname@lri.fr

Functional annotation consists in assigning a biological function to a given protein. It is a crucial task in biology and has various impacts on many fields, including understanding cellular processes and drug designing. In order to be able to share and reuse annotations, biologists and bioinformaticians have developed structured controlled vocabularies that were first simple classifications and then more elaborated ontologies such as the Gene Ontology [1].

In our project, biologists and bioinformaticians collaborators are interested in proteins annotated with two distinct ontologies, such that no protein is annotated with both of them. These ontologies are merely functional hierarchies (Subtilist [2] and FunCat [3]) that share common features: (i) a simple structure with no explicit relationships (subsumption relationships can be deduced from concepts identifiers), (ii) high broadness and small depth, and (iii) variable size.

The system O'Browser [4] we have designed to align functional hierarchies, is based on a weighted combination of matchers as many ontology matching systems [5], with two original characteristics. Indeed, we had to face two issues: (a) a high number of candidates pairs of concepts, and (b) a variable quality of the results of the matchers with respect to the gold standard built by the expert.

As the number of candidates pairs of concepts can be unnecessarily huge, we propose to reduce it by exploiting domain knowledge. For it, we have used **types** (groups of concepts sharing the same semantic context). Concepts that are related to the same field (in our case the same functional genomic field) are assigned to the same type. As an example, the concepts *Utilization of Carbon* and *Synthesis of Glucose* are related to the type *Metabolism*. As in [6], concepts of distinct types will never be mapped (e.g. *Germination* in the context of plants and *Germination* in the context of bacteria). In our approach, an expert manually assigns types to the top concepts of the hierarchies, that represent only a small part of the whole set of concepts of both hierarchies. Types are then spread to all concepts using subsumption relationships. In our experiment, the use of types has allowed to divide the number of candidate pairs by 7. The originality of our contribution is to propose a machine learning strategy to assign types to concepts.

The second issue is about the variable quality of the scores of a given matcher. It has been shown that the good results of a matcher may be spoiled by the scores of other matchers [7, 8]. To address this issue, we would like to give a high weight to a matcher in a combination of matchers only when its results are informative. We claim that the weight of a matcher in a combination should partially depend

on its scores (**adaptive weighting**). As an example, let us consider a string-based matcher that compares concepts from two biological ontologies. If the labels of the concepts are close, the two concepts are likely to be equivalent. On the opposite, distant labels do not indicate necessarily that the concepts are distant. Consequently the weight of the string-based matcher should be high for high scores and weak for low scores.
For each matcher, we define a weighting function which associates a weight to each score of the matcher. Let O_1 (resp. O_2) be the set of concepts of the first (resp. second) ontology and let M_i be a matcher: $O_1 \times O_2 \rightarrow Dom_i$, the weighting function W_i is defined on Dom_i and has $[0, 1]$ as a range. For example, assume that the range of the string-based matcher is $Dom_{String-based} = [0, 1]$. Then a weighting function could be the following simple function: $W_{String-based} : [0, 1] \rightarrow [0, 1]$, where $W_{String-based}(\alpha) = 1$ if $\alpha > 0.5$ and $W_{String-based}(\alpha) = 0.25$ otherwise. Unlike in [9], we allow to associate a strong confidence (and thus a high weight) to low results of a matcher in the case where the score of the matcher is a strong indicator of the absence of equivalence between the considered concepts.

We successfully used types and adaptive weighting to align Subtilist and FunCat and compared the results to the gold standard. O'Browser with adaptive weighting found 80 % of the actual correspondences, while O'Browser with the best classical matcher combination found only 70 % of them.

References

1. The Gene Ontology Consortium: Creating the gene ontology resource: design and implementation. Genome Res. **11** (2001) 1425–1433 http://www.geneontology.org.
2. Moszer, I., Jones, L., Moreira, S., Fabry, C., Danchin, A.: Subtilist: the reference database for the Bacillus subtilis genome. Nucleic Acids Res **30** (2002) 62–5
3. Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Gldener, U., Mannhaupt, G., Mnsterktter, M., Mewes, H.: The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucleic Acids Res. **14**((32)18) (2004) 5539–5545
4. Rance, B., Gibrat, J.F., Froidevaux, C.: An adaptive combination of matchers: application to the mapping of biological ontologies for genome annotation. In: Proc. of the 5th Data Integration in the Life Sciences workshop DILS'09. LNBI 5647 (2009) 113–126
5. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (DE) (2007)
6. Zhang, S., Mork, P., Bodenreider, O., Bernstein, P.A.: Comparing two approaches for aligning representations of anatomy. Artificial Intelligence in Medicine **39**(3) (2007) 227–236
7. Ghazvinian, A., Noy, N.F., Musen, M.A.: Creating mappings for ontologies in biomedicine: Simple methods work. Technical report, Stanford Center for Biomedical Informatics Research (2009)
8. Ontology Alignment Evaluation Initiative: http://www.oaei.ontologymatching.org
9. Mork, P., Seligman, L., Rosenthal, A., Korb, J., Wolf, C.: The harmony integration workbench. J. Data Semantics **11** (2008) 65–93

Parallelization and Distribution Techniques for Ontology Matching in Urban Computing Environments

Axel Tenschert[1], Matthias Assel[1], Alexey Cheptsov[1], Georgina Gallizo[1], Emanuele Della Valle[2], Irene Celino[2]

[1] HLRS – High-Performance Computing Center Stuttgart, University of Stuttgart,
Nobelstraße 19,
70569 Stuttgart, Germany
{tenschert, assel, cheptsov, gallizo}@hlrs.de

[2] CEFRIEL - ICT Institute, Politecnico of Milano,
Via Fucini 2,
20133 Milano, Italy
{emanuelle.dellavalle, irene.celino}@cefriel.it

Abstract. The usage of parallelization and distribution techniques in the field of ontology matching is of high interest for the semantic web community. This work presents an approach for managing the process of extending complex information structures as used in Urban Computing system by means of ontology matching considering parallelization and distribution techniques.

Keywords: Ontology Matching, Semantic Content, Parallelization, Distribution

Ontology Matching through Distribution and Parallelization

Current ontology matching approaches [1] require a high amount of compute resources with the aim to meet the requirements of the matching and merging methods. Hence, several issues have to be considered such as the selection of a suitable ontology, scalability and robustness, matching sequence and identification of the ontology repositories. Approaches for partitioning selected ontologies with the aim to execute matching processes independently from other parts of the ontology are considered to solve this challenge [2]. However, a local ontology matching is a risk for these approaches in terms of scalability and performance issues. Therefore, local ontology matching could be extended by making use of distribution methods as well as parallelization techniques allowing overcoming existing limitations and improving the overall performance.

Within the LarKC project[1], respective techniques for processing large data sets in the research field of the semantic web are investigated and developed. In particular, distribution methods and parallelization techniques are evaluated by executing the matching processes concurrently on distributed and diverse compute resources. A

[1] LarKC (abbr. The Large Knowledge Collider): http://www.larkc.eu

dedicated use case in LarKC deals with the application of these techniques for Urban Computing problems [3].

Common ontology matching algorithms often perform computation intensive operations and thus being considerably time consuming. That poses a number of challenges towards their practical applicability for complex tasks and efficient utilization of the computing architectures that best fit the requirements in order to achieve maximal performance and scalability of the performed operations [4]. Distributed ontology matching enables the use of diverse computing resources, from users' desktop computers to heterogeneous Grid/Cloud infrastructures. Parallelization is the main approach for the effective ontology matching, especially when time characteristics are settled to the point. When thinking of matching several parts of an ontology in parallel in a cluster environment, the matching processes needs to be partitioned. After processing the data, the parts of the ontology have to be merged together again and an extended ontology is generated [5].

Several techniques can be recognized for the parallel implementation of distributed ontology matching.

- Single Code Multiple Data (SCMD workflow)
  In this case the data that is being processed in the code region can be constructed of subsets that have no dependencies between them. The same operation is performed on each of these subsets.
- Multiple Code Single Data (MCSD workflow without conveyer dependencies)
  For this workflow, several different operations are performed on the same dataset. Herewith, no dependencies between processed data sets exist. This is typical for a transformation of one dataset to another one according to rules, which are specific for each subset of the produced data.
- Multiple Code Multiple Data (MCMD workflow)
  This type of workflow is the combination of both previous workflows (SCMD and MSCD).

The presented approach is an effective method to solve the challenge of matching large ontologies in a scalable, robust and timesaving way. Within the LarKC project, these parallelization and distribution techniques for processing semantic data structures are deeply analyzed and further developed.

Acknowledgments. This work has been supported by the LarKC project (http://www.larkc.eu), partly funded by the European Commission's IST activity of the 7th Framework Program. This work expresses only the opinions of the authors.

References

1. Alasoud, A., Haarslev, V., Shiri N.: An empirical comparison of ontology matching techniques. Journal of Information Science 35, 379--397 (2009)
2. Hu W., Cheng G., Zheng D., Zhong X., Qu Y.: The Results of Falcon-AO in the OAEI 2006 Campaign. Ontology Alignment Evaluation Initiative (2006)
3. Kindberg, T., Chalmers, M., Paulos E.: Introduction: Urban Computing. IEEE Pervasive Computing 6, 18--20 (2007)
4. Shvaiko, P., Euzenat, J.: Ten Challenges for Ontology Matching. In: Proceedings of ODBASE, LNCS 5332, pp. 1164--1182, Springer (2008)
5. Dean, J., Ghemawat, S.: Simplified Data Processing on Large Clusters. OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco (2004)

CompositeMatch: Detecting N-Ary Matches in Ontology Alignment*

Kelly Moran[1], Kajal Claypool[1], and Benjamin Hescott[2]

[1] MIT Lincoln Laboratory
{kmoran,claypool}@ll.mit.edu
[2] Tufts University
hescott@cs.tufts.edu

Abstract. The field of ontology alignment still contains numerous unresolved problems, one of which is the accurate identification of composite matches. In this work, we present a context-sensitive ontology alignment algorithm, CompositeMatch, that identifies these matches, along with the typical one-to-one matches, by looking more broadly at the information that a concept's relationships confer. We show that our algorithm can identify composite matches with greater confidence than current tools.

1 Introduction

Numerous ontology alignment algorithms have been developed during recent years to handle the growing challenge of aligning ontologies. While numerous advances have been made, most ontology alignment algorithms are still somewhat inadequate at identifying complex relationship-based matches, a la composite matches. A composite match is defined as a match between a concept c_0 in ontology O and multiple concepts c_0'...c_n' in ontology O'; the reverse; or alternatively between multiple concepts in O and multiple concepts in O'. For example, if an ontology O contains the concept *Name* and an ontology O' contains concepts *FirstName* and *LastName*, then *Name* matches neither *FirstName* nor *LastName* but the composite of the two. In this work, we present *CompositeMatch*, an algorithm that combines linguistic and contextual approaches for semi-automatically discovering both one-to-one and composite matches.

2 The CompositeMatch Algorithm

CompositeMatch is a three-pass algorithm that operates on two input ontologies and produces an alignment file containing matches between them. The **first phase** performs a linguistic match between the ontologies' concepts. The linguistic match assigns a normalized similarity score between 0 and 1.0 to each pair.

* This work was sponsored by the Federal Aviation Administration under Air Force Contract No. FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

In the **second phase**, the most uncertain pairs– collectively referred to as the *grey zone*– are judged on contextual criteria to determine whether they should be accepted as viable matches. The grey zone consists of all conflicting matches– matches that contain the same concept, rendering unclear which match is the true match for the concept– plus matches with a similarity score between the upper and lower thresholds set prior to execution. The second phase defines two rules that serve as a filtering process to increase the scores of contextually similar matches.

The **third and final phase** is a post-processing phase that scours the matches for possible composite matches, looking again at contextual criteria for any indicative information, before outputting the final set of matches to the user.

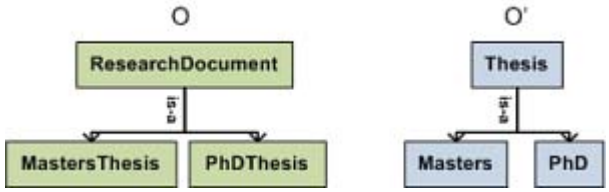


Fig. 1. An m:n composite match

Example 1. Consider the case shown in Figure 1. The first phase finds some similarity between the concept *MastersThesis* in ontology `O` and concepts *Masters* and *Thesis* in ontology `O'`, as well as *PhDThesis* in `O` and *PhD* and *Thesis* in `O'`. Phase 2 compares the parents of these pairs but does not find sufficient contextual similarity to augment the strengths of any of them. In Phase 3, the algorithm finds that the conflicting matches identified earlier result in composite concepts: the concepts *MastersThesis* and *PhDThesis* from `O` form a composite concept that matches the composite concept between *Thesis*, *Masters*, and *PhD* in `O'`.

3 Preliminary Results

We evaluated the performance of CompositeMatch on two tests, the first being the benchmark test from the Ontology Alignment Evaluation Initiative (OAEI) 2008. Because the OAEI benchmark does not account for composite matches, we created a second test: a set of six ontologies, each a modified version of the OAEI benchmark base case, into which composite matches were injected.

We compared the results of CompositeMatch on the OAEI 2008 benchmark to those of a high-performing OAEI 2008 entrant, RiMOM. The mean performance of CompositeMatch on the tests is a precision of .926 and a recall of .557. RiMOM has an overall precision of .939 and a recall of .802. We also considered the subset of all tests not including random or foreign language labels as we believe this subset is a better indicator of how CompositeMatch performs on its intended data sets. CompositeMatch achieves a significantly higher precision and recall on this subset of .996 and .896 respectively, and RiMOM increases slightly to .965 and .967.

While further evaluation is needed, our preliminary results indicate that CompositeMatch correctly identifies each composite match, achieving both a precision and a recall of 1.0, while RiMOM identifies none.

Recommendations for Qualitative Ontology Matching Evaluations

Aliaksandr Autayeu, Vincenzo Maltese, and Pierre Andrews

DISI, University of Trento, Italy

Abstract. This paper suggests appropriate rules to set up ontology matching evaluations and for golden standard construction and use which can significantly improve the quality of the precision and recall measures.

We focus on the problem of evaluating ontology matching techniques [1] which find mappings with *equivalence*, *less general*, *more general* and *disjointness*, and on how to make the evaluation results fairer and more accurate.

The literature discusses the appropriateness and quality of the measures [2], but contains little about evaluation methodology [3]. Closer to us, [4] raises the issue of evaluating non-equivalence links.

Golden standards (GS) are fundamental for evaluating the precision and recall [2]. Typically, hand-made positive (GS^+) and negative (GS^-) golden standards contain links considered correct and incorrect, respectively. Ideally, GS^- complements GS^+, leading to a precise evaluation. Yet, in big datasets annotating all links is impractical and golden standards are often a sample of all node pairs, leading to approximate evaluations [5]. However, most current evaluation campaigns tend to use tiny ontologies, risking biased or poorly significant results.

> *Recommendation 1.* Use large golden standards. Include GS^- for a good approximation of the precision and recall. To be statistically significant, cover in GS^+ and GS^- an adequate portion of all node pairs.

In a sampled GS, results reliability depends on: (a) the portion of the pairs covered; (b) the ratio between GS^+ and GS^- sizes and (c) their quality (see last recommendation).

Most matching tools produce *equivalence*, some also produce *less general* and *more general* relations, but few output *disjointness* [6]. This must be taken into account to correctly compare evaluations. Usually, only the presence of a relation is evaluated, regardless the kind. Moreover, *disjointness* (two completely unrelated nodes) is often confused with *overlap* (two nodes whose intersection is not empty) and both are put in the GS^- [5]. This leads to imprecise results.

> *Recommendation 2.* When presenting evaluation results, specify whether and how the evaluation takes into account the semantic relations kind.

We use the notion of redundancy [7] to judge the quality of a golden standard. We use the **Min(mapping)** function to remove redundant links (producing the *minimized mapping*) and the **Max(mapping)** function to add all

redundant links (producing the *maximized mapping*). Following [7] and staying within lightweight ontologies [8] guarantees that the maximized set is always finite and thus precision and recall can always be computed. The table below presents the measures obtained in our experiments with SMatch on three different datasets (see [6] for details). Comparing the measures obtained with the maximized versions (max) with the measures obtained with the original versions (res), one can notice that the performance of the algorithm is on average better than expected. In [6] we explain why comparing the minimized versions is not meaninful and we conclude that:

> *Recommendation 3.* To obtain accurate measures it is fundamental to maximize both the golden standard and the matching result.

Dataset pair	Precision, %			Recall, %		
	min	res	max	min	res	max
101/304	32.47	9.75	69.67	86.21	93.10	92.79
Topia/Icon	16.87	4.86	45.42	10.73	20.00	42.11
Source/Target	74.88	52.03	48.40	10.35	40.74	53.30

Maximizing a golden standard can also reveal unexpected problems and inconsistencies. For instance, we discovered that in TaxME2 [5] $|GS^+ \cap GS^-| = 2$ and $|Max(GS^+) \cap Max(GS^-)| = 2187$. In future work we will explore how the size of the golden standard influences the evaluation and how large should be the part covered by GS^+ and GS^-, as well as describe methodology for evaluating rich mappings by supporting our recommendations with experimental results.

References

1. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. JoDS **4** (2005) 146–171
2. David, J., Euzenat, J.: On fixing semantic alignment evaluation measures. In: Proc. of the 3rd Ontology Matching Workshop. (2008)
3. Noy, N.F., Musen, M.A.: Evaluating ontology-mapping tools: Requirements and experience. In: Proc. of OntoWeb-SIG3 Workshop. (2002) 1–14
4. Sabou, M., Gracia, J.: Spider: Bringing non-equivalence mappings to OAEI. In: Proc. of the 3rd Ontology Matching Workshop. (2008)
5. Giunchiglia, F., Yatskevich, M., Avesani, P., Shvaiko, P.: A large dataset for the evaluation of ontology matching systems. KERJ **24** (2008) 137–157
6. Autayeu, A., Maltese, V., Andrews, P.: Best practices for ontology matching tools evaluation. Technical report, University of Trento, DISI (2009)
7. Giunchiglia, F., Maltese, V., Autayeu, A.: Computing minimal mappings. In: Proc. of the 4th Ontology Matching Workshop. (2009)
8. Giunchiglia, F., Marchese, M., Zaihrayeu, I.: Encoding classifications into lightweight ontologies. JoDS **8** (2007) 57–81

Implementing Semantic Precision and Recall

Daniel Fleischhacker and Heiner Stuckenschmidt

University of Mannheim, Mannheim, Germany
dfleisch@mail.uni-mannheim.de, heiner@informatik.uni-mannheim.de

1 Introduction

The systematic evaluation of ontology alignments still faces a number of problems. One is the argued inadequacy of traditional quality measures adopted from the field of information retrieval. In previous work, Euzenat and others have proposed notions of semantic precision and recall that are supposed to better reflect the true quality of an alignment by considering the deductive closure of a mapping rather than the explicitly stated correspondences. So far, these measures have been mostly investigated in theory. In this paper, we present the first implementation of a restricted version of semantic precision and recall as well as experiments in using it, we conducted on the results of the 2008 OAEI campaign.

2 Restricted Semantic Precision and Recall

In this work, we treat alignments as sets of correspondences whereas correspondences give a relation between two entities from different ontologies. To evaluate alignments, we use the notion of aligned ontologies. An aligned ontology is made of the two ontologies which are referenced by an alignment and the correspondences contained in this alignment added into the aligned ontology as axioms. To convert correspondences into axioms, we use semantics as the natural and pragmatic semantics given by Meilicke and Stuckenschmidt [3]. The basis of our work is the work of Euzenat [2] which we adapted to our different understanding of alignment semantics. The basic notion given by Euzenat and used here is the notion of α-consequences. These are correspondences which are implied by an aligned ontology given specific semantics. For ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, a corresponding alignment A and reductionistic semantics S, we say $A \models^S_{\mathcal{O}_1,\mathcal{O}_2} c$ if c is an α-consequence.

Applying this definition to complete alignments instead of single correspondences, we get the closure of an alignment which resembles the sets of α-correspondences used by Euzenat. For given ontologies $\mathcal{O}_1$, $\mathcal{O}_2$ and a reductionistic semantics S the closure Cn of an alignment A is given by $\mathrm{Cn}^S_{\mathcal{O}_1,\mathcal{O}_2}(A) = \{c \mid A \models^S_{\mathcal{O}_1,\mathcal{O}_2} c\}$.

We introduce a restricted variant of ideal semantic precision and recall which does not suffer from the problems of the ideal semantic precision and recall mentioned by Euzenat [2] and also prevent problems examined by David and Euzenat [1]. For this purpose, we call alignments non-complex if they contain only correspondences whose entities refer to single atomic concepts of the ontologies.

Definition 1 (Restricted Semantic Precision and Recall). *Given consistent ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, two non-complex alignments between these two ontologies, namely the reference alignment R and the alignment A which is to be evaluated, and a reductionistic semantics S. Further, let the aligned ontologies of the two ontologies with A resp. R be consistent. Restricted semantic precision and recall are defined as*

$$\mathrm{P}_r(A,R) = \frac{|\mathrm{Cn}^S_{\mathcal{O}_1,\mathcal{O}_2}(A) \cap \mathrm{Cn}^S_{\mathcal{O}_1,\mathcal{O}_2}(R)|}{|\mathrm{Cn}^S_{\mathcal{O}_1,\mathcal{O}_2}(A)|} \textit{ resp. } \mathrm{R}_r(A,R) = \frac{|\mathrm{Cn}^S_{\mathcal{O}_1,\mathcal{O}_2}(A) \cap \mathrm{Cn}^S_{\mathcal{O}_1,\mathcal{O}_2}(R)|}{|\mathrm{Cn}^S_{\mathcal{O}_1,\mathcal{O}_2}(R)|}$$

3 First Results

Matcher	Semantics	0.2		0.5		0.7	
		P	R	P	R	P	R
ASMOV	*none*	0.42	0.42	0.7	0.18	0.81	0.09
	natural	0.39	0.69	0.81	0.26	1.0	0.15
	pragmatic	0.49	0.74	0.85	0.23	1.0	0.13
DSSim	*none*	0.49	0.52	0.49	0.52	0.49	0.52
	natural	0.15	0.83	0.15	0.83	0.15	0.83
	pragmatic	0.23	0.88	0.23	0.88	0.23	0.88
Lily	*none*	0.5	0.36	0.54	0.21	0.66	0.07
	natural	0.45	0.46	0.65	0.24	0.74	0.09
	pragmatic	0.48	0.51	0.66	0.22	0.65	0.07

Table 1. Aggregated precision (P) and recall (R) results of conference test set comparing classical precision and recall (no semantics), natural and pragmatic precision and recall; top-most line gives minimum confidence value (threshold) to consider a correspondence

We applied the measures to two different test sets taken from the OAEI test sets. In the following, we only present the results generated for the conference test set of the OAEI 2008. We evaluated the alignments provided by the developers of the ontology matchers. Aggregated results for the conference set are presented in Table 1. The aggregation is done using the average of all values for a specific measure which are neither an error entry nor have the value „nan".

4 Conclusion

Our results show that taking the semantics of the model into account can make a difference in judging the quality of matching systems not only in theory but also in practice. So far, this effect is rather limited, which is mainly due to the fact that most generated alignments as well as reference alignments only consist of equivalence statements. It is clear, however, that future work will also strongly focus on generating mappings other than equivalence mappings. Further, there is an ongoing effort to extend existing reference alignments with subsumption correspondences. In such an extended setting, the effect of the semantic measures will be even higher and our system will show its real potential for improving ontology mapping evaluation.

References

1. Jérôme David and Jérôme Euzenat. On fixing semantic alignment evaluation measures. In *Proceedings of the ISWC 2008 Workshop on Ontology Matching*, pages 25–36, 2008.
2. Jérôme Euzenat. Semantic precision and recall for ontology alignment evaluation. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 348–353, 2007.
3. Christian Meilicke and Heiner Stuckenschmidt. An Efficient Method for Computing a Local Optimal Alignment Diagnosis. Technical report, University of Mannheim, 2009.

Learning to Map Ontologies with Neural Network

Yefei Peng, Paul Munro
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15206 USA
{ypeng, pmunro}@mail.sis.pitt.edu

Ming Mao
SAP Labs
Palo Alto, CA 94304 USA
Ming.mao@sap.com

Abstract. In this paper the authors applied the idea of training multiple tasks simultaneously on a partially shared feed forward network to domain of ontology mapping. A "cross training" mechanism was used to specify corresponding nodes between the two ontologies. By examining output of one network in response to stimulus from the other network, we can test if the network can learn the correspondence that was not cross-trained. Two kinds of studies on ontology mapping were conducted. The result shows the network can fill in the missing mappings between ontologies with sufficient training data.

Keywords: neural network, shared weights, transfer, analogy, ontology mapping

An early implementation of IENN appeared at Munro's work [2], which used feedforward network with two hidden layers and trained on three simple analogous tasks: three squares with different orientation. In this study, we use partially shared network architecture [3] [4]. It should be noted that the partially shared network architecture used here is virtually identical to the network used in Hinton's [1] classic "family trees" example. The network in that paper also had independent inputs and shared hidden units, but only briefly addresses the notion of generalization.

The ontologies used in our experiment were Ontology A and B shown in Figure 1. There are four types of relationship: identity, parent, child, and sibling. So there are 4 nodes in S_{in}. Training in Net_A include all possible training data in Ontology A, i.e. possible combinations of 6 nodes and 4 relationships. The same for Net_B.

The network is cross trained on the following pairs: (r, R), (a, A), (b, B), (c, C) and (d, D).

Totally 100 trials were performed. In each trial, networks were initialized by setting the weights to small random values from a uniform distribution. The network was trained with two vertical training tasks (Net_A and Net_B), and two cross training tasks(Net_{AB} and Net_{BA}).

One training cycle of the networks is

1) randomly train a record for Net_A
2) randomly train a record for Net_B
3) with a probability train a record for Net_{AB} and the same record for Net_{BA}.

The probability of cross training is 0.6.

After each trial, cross-testing was performed for A:1, B:2, B:3, and B:4. "self" relationship was used during cross-testing.

In 100 trials, 93 of them yield correct mapping for A:1maps to B:2. The accuracy is 93%. There is no doubt that B:2's correct mapping should be A:1, which is (Car, Car). But B:3 (Luxury Car) and B:4 (Family Car) do not have exact correspondences in ontology A, since B:3 and B:4 are on the additional layer of ontology B compared to ontology A. They can either go up one layer and map to A:1, or go down one layer and map to A:C and A:D. So here the correct mapping will be (A:1, B:3), (A:C, B:3); (A:1, B:4), (A:D, B:4). Totally the four correct cases contain 75 trials in 100 trials. The accuracy is 75%.

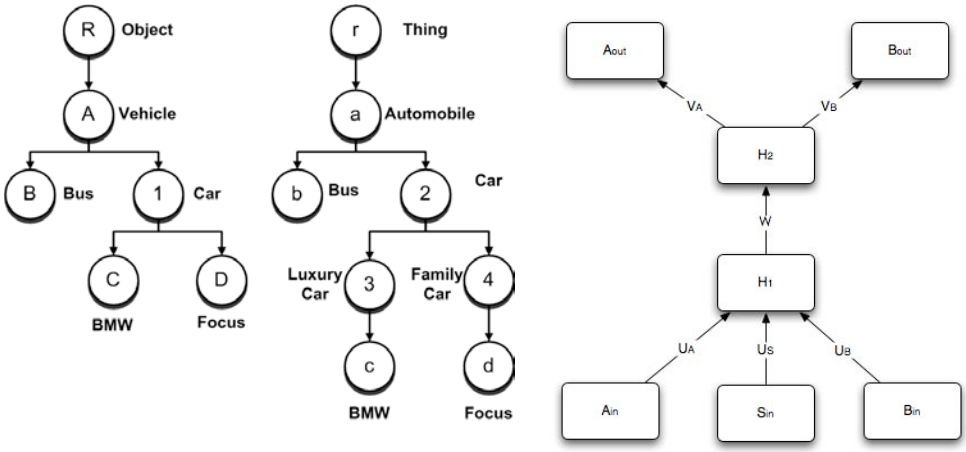


Figure 1. Left:Two sample ontologies about vehicle. Right: Network architecture

The ability to establish correspondences between similar situations is fundamental to intelligent behavior. Here, a network has been introduced that can identify corresponding items in analogous spaces. A key feature of this approach is that there is no need for the tasks to use a common representation. Essentially the first hidden layer provides a common representational scheme for all the input spaces.

In our approach, only structure information is used for ontology mapping. Normally in ontology mapping methods, textual information plays an important role. Future work will be to include textual information in our neural network. For example, training pairs could be from high confident mappings from textual information.

1. Hinton, G. (1986). Learning distributed representations of concepts. In Proceedings of the Eighth Annual Conference of the Cognitive Science Society, pages 1-12, Amherst, Lawrence Erlbaum, Hillsdale.
2. Munro, P. (1996) Shared network resources and shared task properties. In: Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society. Mahwah NJ: Erlbaum
3. Munro, P. (2008) Learning Structurally Analogous Tasks. In: Proceedings of the Eighteenth Conference of Artificial Neural Networks. Prague, Czech Republic
4. Peng, Yefei and Munro, P. " Learning Mappings with Neural Network" , In the Proceedings of the 2009 International Conference on Artificial Intelligence

Matching Natural Language Data with Ontologies

Johannes Heinecke

Orange Labs, F-22307 Lannion cedex – France
johannes.heinecke@orange-ftgroup.com

Abstract. Ontologies and natural languages are complementary. Whereas ontologies are used to model knowledge formally, natural language is primarily used by users to communicate with ontology based systems. In order to transform information or queries in natural language into valid ontological expressions, the meaning of natural language entities have to be matched with the given ontologies. In contrast to pure ontology matching, the matching with natural language data poses some problems linked to their ambiguities (synonymy, homonymy/polysemy, redundancy, to name but a few).

1 Introduction, context and related work

In the context of the Semantic Web, the interfacing of ontological representation and natural language is an important issue. Since much information on the Web exists (only) in textual form, the usage of this information in ontology based tools is not possible unless these texts are made accessible or comprehensible by such tools. This means that texts and user queries have to be "translated" into an ontological representation language such as the W3C languages RDF/RDFS and OWL.

The need for the work described here came from the aceMedia project (*http://www.acemedia.org/*) [1,2] (cf. also [3]). In this project, the two tasks are

transforming textual annotations of multimedia contents into an ontological representation (based on an existing ontology) in order to make them available for a knowledge-base; and translating English and French user queries into an ontological query language (in our case SPARQL). The matching of linguistic data (lexicons, thesauri) with ontologies is similar but not identical to ontology matching or ontology alignment, i.e. trying to find corresponding classes of ontology *A* in ontology *B* [4]. Different methods of matching are discussed in detail by [5, p. 65]. Following this classification the present approach can be considered being terminological and linguistic, since we use relationships found in the lexicon (via a semantic thesaurus, [6]) and the taxonomic hierarchies of both, the lexical semantic data and the ontologies notably for the disambiguation of polysemous words. Similar work describe [7] and [8]. In contrast to their results we do not have a classification at hand.

2 Linguistic-ontology matching

Apart from the ontologies, the matching requires a complete lexicon of the language used to label or describe the ontological classes and properties (= entities). Our lexicon is lso linked to a semantic thesaurus. The ontologies, on the other hand, usually have non-ambiguous entity labels (like *http: //www.acemedia.org/ontos/tennis#Player*[1]) or a comment, explaining the entity. This is especially necessary if the entity labels are not self-explanatory like

[1] We shorten name spaces like *http://www.acemedia.org/ontos/tennis#* to "*tennis:*" etc.

tennis:C12 (a fortunately rare case). Further, the semantic thesaurus contains a thematic hierarchy of all semantic concepts to help disambiguation. These are grouped into 880 themes which in turn are organized in 80 domains. Domains are divided into about 10 macro-domains. The matching itself comprises several steps (cf. fig. 1). Apart from a (more or less manual) preparation in order to correct possible labeling errors in the ontologies, the other steps do not need any intervention: (a) extracting the "ontological context" of entities and assigning eventual reformulations of entity labels; (b) natural language processing passes: detecting meanings for classes using their ontological context (direct sub-classes); (c) and for properties using their ontological context (domain and range classes); (d) determining the application depending synonyms and co-hyponyms; (e) adding the ontological hierarchy to the semantic taxonomy; (f) creating semantic transformation rules for "complex class labels"[2]; (g) creating transformation rules for the creation of ontological representation (from semantic graphs. Synonyms (defined in our multilingual thesaurus, [9]) are all matched onto the same ontological class (e.g. "river", "stream", "creek" etc. → *holidays:River*). If a class has no sub-classes, we also match the co-hyponyms of the label to the class (e.g. in our case "car", "bus", "truck", "motorbike" ... → *general:Vehicle*. The resulting linguistic data is successfully used the aceMedia prototype, similarly produced data is used in an industrial application to create and access ontological based information from/via natural language.

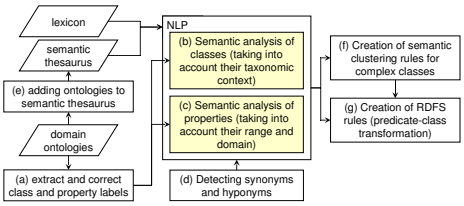


Fig. 1. linguistic-ontological matching

New perspectives are offered by structured semantic data which is getting more and more available. Databases like Wikipedia (especially the categorization schema used within) or RDF or ontology based information systems like DBpedia or freebase[3] (both initialized by Wikipedia contents) will help to improve the linking of natural languages and formally modeled ontologies.

References

1. Heinecke, J.: Génération automatique des représentations ontologiques. In: TALN. (2006) 502–511
2. Dasiopoulou, S., Heinecke, J., Saathoff, C., Strintzis, M.G.: Multimedia reasoning with natural language support. In: IEEE-ICSC. (2007) 413–420
3. Heinecke, J., Toumani, F.: A Natural Language Mediation System for E-Commerce applications. In: Workshop HLT for the Semantic Web and Web Services. ISWC. (2003) 39–50
4. Ehrig, M., Staab, S.: QOM - quick ontology mapping. In: ISWC. (2004) 683–697
5. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)
6. Heinecke, J., Smits, G., Chardenon, C., Guimier De Neef, E., Maillebuau, E., Boualem, M.: TiLT : plate-forme pour le traitement automatique des langues naturelles. TAL **49:2** (2008)
7. Giunchiglia, F., Marchese, M., Zaihrayeu, I.: Encoding classifications into lightweight ontologies. Journal of Data Semantics VIII (2007) 57–81
8. Reiter, N., Hartung, M., Frank, A.: A resource-poor approach for linking ontology classes to Wikipedia articles. In: STEP. (2008) 381–387
9. Chagnoux, M., Heinecke, J.: Aligner ontologies et langues naturelles. gérer la synonymie. In: Plateforme AFIA, Grenoble (2007) 87–94

[2] Labels which use multi-word expressions like *tennis:ExhibitionMatch* instead of simple words.

[3] *http://dbpedia.org/*, *http://freebase.com/*

Reducing polysemy in WordNet

Kanjana Jiamjitvanich, Mikalai Yatskevich

Department of Information and Communication Technology,
University of Trento, Italy
kanjana@disi.unitn.it, yatskevi@disi.unitn.it

1 WordNet

WordNet [4] is the lexical database for English language. A synset is a WordNet structure for storing senses of the words. Synset contains a set of synonym words and their brief description called gloss. For example, *well*, *wellspring* and *fountainhead* have the same meaning according to WordNet, so these three words are grouped in to one synset which is explained by a gloss "*an abundant source*".

A known problem of WordNet is that it is too fine-grained in its sense definitions. For instance, it does not distinguish between homographs (words that have the same spelling and different meanings) and polysemes (words that have related meanings). We propose to distinguish only between polysemes within WordNet while merging all homograph synsets. The ultimate goal is to compute a more coarse-grained version of linguistic database.

2 Meta matcher

Meta matcher is designed as a WordNet matcher, i.e., a matcher that is effective in matching WordNet with itself. It utilizes extensible set of element level matchers (see [1] for extensive discussion) and combines their results in hybrid manner, i.e., the final score is computed from the scores of independently executed matchers.

We implemented three element level matchers.

WordNet relation matcher (WNR). WNR takes two senses as an input and obtains two sets of senses connected to input senses by a given relation. Then these two sets are compared exploiting well-known Dice coefficient formula.

Part of speech context (POSC). POSC matcher exploits part of speech (POS) and sense tagged corpora for similarity computation. In particular, for each WordNet sense occurrence within corpora a set POS tags in the immediate vicinity of sense is memorized. Given multiple occurrence of a sense within corpora each sense is associated with a set of POS contexts. Then, the similarity between two senses is computed as set similarity between sets of POS contexts associated with them.

Inverted sense index inexact (ISII). ISII matcher exploits sense tagged WordNet 3.0 glosses for similarity computation. In particular, for each WordNet sense occurrence within sense tagged glosses, the synset of a tagged gloss is memorized. Than, senses are compared by comparing sets of synsets associated with them. We compare synsets exploiting well known Resnik similarity measure [6].

Matching process is organized in two steps.

2.1 Element level matchers threshold learning

The necessary prerequisite for this step is a training dataset or (a part of) the matching task for which human alignment *H* is known. All element level matchers then are executed on the training dataset, i.e., we obtain complete set of correspondences *M* for all matchers. Then the threshold learning procedure is executed. It performs exhaustive search through all threshold combinations for all element level matchers. Thus, we can select threshold that maximizes a given matching quality metric, e.g., Recall.

In the case of several matchers system result set *S* is obtained from their results through a combination strategy, namely a function that takes matchers results in input and produces a binary decision of whether the given correspondence holds. In this paper we used union of all matchers results as a combination strategy, i.e., if a given correspondence is returned by at least one matcher it is included in *S*.

2.2 Hybrid matching

On this step meta matcher is executed on testing dataset. Element level matchers results are combined using thresholds and the combination strategy exploited in the previous step. For union combination strategy positive result is produced only if confidence score, as computed by element level matchers, is higher than threshold learned on the previous step.

3 Evaluation results

We used a dataset exploited in SemEval[1] evaluation. The dataset contains 1108 nouns, 591 verbs, 262 adjectives and 208 adverbs. We split it into two equal parts: training and testing datasets.

We compared results of meta matcher with 3 other sense merging methods. In particular, we re-implemented sense merging algorithm [2], Genclust algorithm [5] and MiMo algorithm [3]. Meta matcher outperforms the other methods in terms of F-Measure.

References

1. F. Giunchiglia and M. Yatskevich. Element level semantic matching. In The Semantic Web: ISWC 2004: Third International Semantic Web Conference: Proceedings, 2004.
2. W. Meng, R. Hemayati, and C. Yu. Semantic-based grouping of search engine results using wordnet. In 9th Asia-Pacific Web Conference (AP-Web/WAIM'07), 2007.
3. R. Mihalcea and D. Moldovan. Automatic generation of a coarse-grained wordnet. In NAACL Workshop on WordNet, 2001.
4. G. Miller. WordNet: An electronic Lexical Database. MIT Press, 1998.
5. W. Peters, I. Peters, and P. Vossen. Automatic sense clustering in eurowordnet. In Proceedings of LREC'1998, 1998.
6. P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research, 11:95{130, 1999.

[1] http://lcl.di.uniroma1.it/coarse-grained-aw/index.html