Ontology Matching

OM-2010

Proceedings of the ISWC Workshop

Introduction

Ontology matching[1] 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 two goals:

- To bring together leaders from *academia*, *industry* and *user institutions* to assess how academic advances are addressing real-world requirements. The workshop strives to improve academic awareness of industrial and final user needs, and therefore, direct research towards those needs. Simultaneously, the workshop serves to inform industry and user representatives about existing research efforts that may meet their requirements. The workshop also investigates 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) 2010 campaign[2]. The particular focus of this year's OAEI campaign is on real-world specific matching tasks involving, e.g., biomedical ontologies and open linked data. Thus, the ontology matching evaluation initiative itself provides a solid ground for discussion of how well the current approaches are meeting business needs.

We received 29 submissions for the technical track of the workshop. The program committee selected 7 submissions for oral presentation and 13 submissions for poster presentation. 15 matching systems participated in this year's OAEI campaign. Further information about the Ontology Matching workshop can be found at: `http://om2010.ontologymatching.org/`.

---

[1] `http://www.ontologymatching.org/`

[2] `http://oaei.ontologymatching.org/2010`

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)[3] initiative of the European Network of the Living Labs[4] at Informatica Trentina SpA[5], the EU SEALS (Semantic Evaluation at Large Scale)[6] project and the Semantic Valley[7] initiative.

Pavel Shvaiko
Jérôme Euzenat
Fausto Giunchiglia
Heiner Stuckenschmidt
Ming Mao
Isabel Cruz

November 2010

---

[3] `http://www.taslab.eu`
[4] `http://www.openlivinglabs.eu`
[5] `http://www.infotn.it`
[6] `http://www.seals-project.eu`
[7] `http://www.semanticvalley.org/index_eng.htm`

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
Ming Mao, SAP Labs, USA
Isabel Cruz, The University of Illinois at Chicago, USA

Program Committee

Paolo Besana, Université de Rennes 1, France
Olivier Bodenreider, National Library of Medicine, USA
Marco Combetto, Informatica Trentina, Italy
Jérôme David, Université Pierre Mendes-France, INRIA & LIG, France
AnHai Doan, University of Wisconsin and Kosmix Corp., USA
Alfio Ferrara, University of Milan, Italy
Tom Heath, Talis, UK
Wei Hu, Nanjing University, China
Ryutaro Ichise, National Institute of Informatics, Japan
Antoine Isaac, Vrije Universiteit Amsterdam & Europeana, Netherlands
Krzysztof Janowicz, Pennsylvania State University, USA
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 La Sapienza, Italy
Juanzi Li, Tsinghua University, China
Augusto Mabboni, Business Process Engineering, Italy
Vincenzo Maltese, University of Trento, Italy
Fiona McNeill, University of Edinburgh, UK
Christian Meilicke, University of Mannheim, Germany
Luca Mion, Informatica Trentina, Italy
Peter Mork, The MITRE Corporation, USA
Filippo Nardelli, Cogito, Italy
Natasha Noy, Stanford University, USA
Leo Obrst, The MITRE Corporation, USA
Yefei Peng, Google, USA
Erhard Rahm, University of Leipzig, Germany
François Scharffe, INRIA & LIG, France
Luciano Serafini, Fondazione Bruno Kessler - IRST, Italy
Kavitha Srinivas, IBM, USA

Umberto Straccia, ISTI-C.N.R., Italy
Andrei Tamilin, Fondazione Bruno Kessler - IRST, Italy
Cássia Trojahn dos Santos, INRIA & LIG, France
Lorenzino Vaccari, European Commission - Joint Research Center, Italy
Ludger van Elst, DFKI, Germany
Yannis Velegrakis, University of Trento, Italy
Shenghui Wang, Vrije Universiteit Amsterdam, Netherlands
Baoshi Yan, Bosch Research, USA
Rui Zhang, Jilin University, China
Songmao Zhang, Chinese Academy of Sciences, China

Additional Reviewers

Zhongli Ding, Google, USA
Eugenio Di Sciascio, Politecnico di Bari, Italy
Songyun Duan, IBM, USA
Jörn Hees, DFKI, Germany
Anika Gross, University of Leipzig, Germany
Nico Lavarini, Cogito, Italy
Sabine Massmann, University of Leipzig, Germany
Salvatore Raunich, University of Leipzig, Germany
Zhichun Wang, Tsinghua University, China

Table of Contents

PART 1 - Technical Papers

Linguistic analysis for complex ontology matching
Dominique Ritze, Johanna Völker,
Christian Meilicke and Ondřej Šváb-Zamazal 1

Lost in translation? Empirical analysis of mapping compositions
for large ontologies
Anna Tordai, Amir Ghazvinian, Jacco van Ossenbruggen,
Mark Musen and Natasha Noy 13

Chinese whispers and connected alignments
Oliver Kutz, Immanuel Normann, Till Mossakowski and Dirk Walther 25

Consistency-driven argumentation for alignment agreement
Cássia Trojahn and Jérôme Euzenat 37

Alignment based measure of the distance between
potentially common parts of lightweight ontologies
Ammar Mechouche, Nathalie Abadie and Sébastien Mustière 49

Mapping the central LOD ontologies to PROTON upper-level ontology
Mariana Damova, Atanas Kiryakov, Kiril Simov and Svetoslav Petrov 61

Ontology alignment in the cloud
Jürgen Bock, Alexander Lenk and Carsten Dänschel 73

PART 2 - OAEI Papers

First results of the Ontology Alignment Evaluation Initiative 2010
Jérôme Euzenat, Alfio Ferrara, Christian Meilicke, Juan Pane,
François Scharffe, Pavel Shvaiko, Heiner Stuckenschmidt,
Ondřej Šváb-Zamazal, Vojtěch Svátek, and Cássia Trojahn dos Santos 85

Using AgreementMaker to align ontologies for OAEI 2010:
Isabel Cruz, Cosmin Stroe, Michele Caci, Federico Caimi,
Matteo Palmonari, Flavio Palandri Antonelli and Ulas C. Keles 118

ASMOV: results for OAEI 2010
Yves R. Jean-Mary, E. Patrick Shironoshita and Mansur R. Kabuka 126

BLOOMS on AgreementMaker: results for OAEI 2010
Catia Pesquita, Cosmin Stroe, Isabel Cruz and Francisco Couto 134

CODI: Combinatorial Optimization for Data Integration:
results for OAEI 2010
Jan Noessner and Mathias Niepert 142

Eff2Match results for OAEI 2010
Watson Wei Khong Chua and Jung-Jae Kim 150

ObjectCoref & Falcon-AO: results for OAEI 2010
Wei Hu, Jianfeng Chen, Gong Cheng and Yuzhong Qu 158

An integrated matching system GeRoMeSuite and SMB:
results for OAEI 2010
Christoph Quix, Avigdor Gal, Tomer Sagi and David Kensche 166

LN2R a knowledge based reference reconciliation system:
OAEI 2010 results
Fatiha Saïs, Nobal Niraula, Nathalie Pernelle and
Marie-Christine Rousset 172

MapPSO results for OAEI 2010
Jürgen Bock 180

Results of NBJLM for OAEI 2010
Song Wang, Gang Wang and Xiaoguang Liu 187

RiMOM results for OAEI 2010
Zhichun Wang, Xiao Zhang, Lei Hou, Yue Zhao, Juanzi Li,
Yu Qi and Jie Tang 195

Alignment results of SOBOM for OAEI 2010
Peigang Xu, Yadong Wang, Liang Cheng and Tianyi Zang 203

TaxoMap alignment and refinement modules: results for OAEI 2010
Fayçal Hamdi, Brigitte Safar, Nobal Niraula and Chantal Reynaud 212

PART 3 - Posters

Towards a UMLS-based silver standard for matching biomedical ontologies
Ernesto Jiménez-Ruiz, Bernardo Cuenca Grau,
Ian Horrocks and Rafael Berlanga 220

From French EHR to NCI ontology via UMLS
Paolo Besana, Marc Cuggia, Oussama Zekri,
Annabel Bourde and Anita Burgun 222

From mappings to modules: using mappings
to identify domain-specific modules in large ontologies
Amir Ghazvinian, Natasha Noy and Mark Musen 224

Harnessing the power of folksonomies for formal
ontology matching on-the-fly
Theodosia Togia, Fiona McNeill and Alan Bundy 226

Concept abduction for semantic matchmaking
in distributed and modular ontologies
Viet-Hoang Vu and Nhan Le-Thanh 228

LingNet: networking linguistic and terminological ontologies
Wim Peters 230

Aggregation of similarity measures in ontology matching
Lihua Zhao and Ryutaro Ichise 232

Using concept and structure similarities for ontology integration
Xiulei Liu, Payam Barnaghi, Klaus Moessner and Jianxin Liao 234

Flexible bootstrapping-based ontology alignment
Prateek Jain, Pascal Hitzler and Amit Sheth 236

Semantic matching of ontologies
Christoph Quix, Marko Pascan, Pratanu Roy and David Kensche 238

Ontology mapping neural network: an approach
to learning and inferring correspondences among ontologies
Yefei Peng, Paul Munro and Ming Mao 240

Towards tailored domain ontologies
Cheikh Niang, Béatrice Bouchou and Moussa Lo 242

Crowd sourcing through social gaming for community driven
ontology engineering, results and observations
Alloy Martin Chua, Roland Christian Chua, Arthur Vincent Dychiching,
Tinmon Ang, Jose Lloyd Espiritu, Nathalie Rose Lim and
Danny Cheng 244

Linguistic Analysis for Complex Ontology Matching

Dominique Ritze[1], Johanna Völker[1], Christian Meilicke[1], and Ondřej Šváb-Zamazal[2]

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

Abstract. Current ontology matching techniques focus on detecting correspondences between atomic concepts and properties. Nevertheless, it is necessary and possible to detect correspondences between complex concept or property descriptions. In this paper, we demonstrate how complex matching can benefit from natural language processing techniques, and propose an enriched set of correspondence patterns leveraging linguistic matching conditions. After elaborating on the integration of methods for the linguistic analysis of textual labels with an existing framework for detecting complex correspondences, we present the results of an experimental evaluation on an OAEI dataset. The results of our experiments indicate a large increase of precision as compared to the original approach, which was based on similarity measures and thresholds.

1 Introduction

Ontology matching can be considered one of the key technologies for efficient knowledge exchange and the successful realization of the Semantic Web. Bridging the gap between different terminological representations is an indispensable requirement for a large variety of tools and technologies including, for example, distributed reasoning, instance migration, and query rewriting in distributed environments.

In the past, ontology matching was commonly considered the task of detecting similar or equivalent concepts and properties in two ontologies. However, this view on ontology matching seems too narrow for many application scenarios, and new requirements have motivated several extensions to the original task definition. Among the challenges of the last Ontology Alignment Evaluation Initiative (OAEI) [4], for example, we find the task of instance matching as well as a track that aims at the generation of correspondences expressing subsumption (instead of equivalence) between concepts.

In our work, we suggest to extend the classical way of ontology matching in a different direction – the generation of correspondences between complex concept and property descriptions. We refer to these correspondences as *complex correspondences* and call the process of generating them as *complex ontology matching*. Our work is motivated by the insight that equivalence or even subsumption correspondences between atomic entities are often not applicable or not expressive enough to capture important dependencies between ontological entities.

This paper is based on our previous approach [11] which we found to have several disadvantages, including problems related to the precision of the patterns as well as the prerequisite of a reference alignment as additional input. In order to address these issues, we modified the original approach in the following way:

- A partial reference alignment is no longer required as input. We generate the alignment in a preprocessing step and use it as an anchor alignment later on.
- The complete logic required to express the conditions for generating correspondences is now described declaratively by means of XML. The XML-based specification of the matching conditions is interpreted and executed by our tool, while the concrete implementation remains transparent to the user.
- We changed the output format of our system so that it adheres to the *Expressive Declarative Ontology Alignment Language* (EDOAL) for complex correspondences, that is supported by the alignment API [3].
- We extended our approach by various methods for the linguistic analysis of concept or property labels. According to our experiments, the appropriate use of these methods results in a significantly increased precision.

The last point refers to the most essential contribution of this paper. In our previous approach, many matching conditions included similarity thresholds. Thanks to linguistic methods, we can now avoid the need for finding appropriate thresholds. In the remainder of this paper, we show that the use of these methods significantly improves the quality of complex matching. In particular, we find that the linguistic analysis enables us to achieve a significantly higher precision with respect to the previously detected [11] pattern instantiations. Moreover, we present a new correspondence pattern leveraging linguistic matching conditions and illustrate the advantages of the new approach by means of concrete examples.

Our paper is structured as follows. In Section 2 we describe our approach to complex matching. We introduce the terminology we adopt, and sketch the core elements of our algorithm. Section 3 discusses the related work, whereas in Section 4, we describe the linguistic methods that we apply to detect non-trivial semantic relations between concepts and properties. The pattern-specific matching conditions, which constitute the heart of our approach, are presented in Section 5. In Section 6, we report on our evaluation experiments, before concluding in Section 7.

2 Approach

Now we introduce the basics of our approach and explain the terminology we use in the subsequent sections. First of all, we adopt and slightly simplify the generic terminology defined in [6]. Thus, we understand an *alignment* between two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ as a set of correspondences. A *correspondence* is a triple $\langle X, Y, Z \rangle$ where X is an entity from $\mathcal{O}_1$, Y is an entity from $\mathcal{O}_2$ and R is a relation such as equivalence or subsumption. Whenever it is required to refer to the origin of a specific concept, we write $C_{\#i}$ to indicate that C belongs to $\mathcal{O}_i$.

State-of-the-art ontology matching techniques are bound to detect correspondences as $\langle \mathit{Paper}, \mathit{Article}, \equiv \rangle$ or $\langle \mathit{writes}, \mathit{writesPaper}, \subseteq \rangle$. In the following we describe an approach that allows to detect correspondences where X and Y are complex concept or property descriptions.

Remember that the power of description logic originates from the ability to build complex concept and property descriptions from atomic ones, i.e. from concept and property names. The following listing shows some of the different ways to construct complex descriptions in description logics (here at the example of $\mathcal{SHOIN}$).

$\neg C$	*(atomic negation)*	$\forall P.C$	*(value restriction)*
$B \sqcap C$	*(conjunction)*	$\exists_{\leq n} P$	*(at least restriction)*
$B \sqcup C$	*(disjunction)*	$\exists_{\geq n} P$	*(at most restriction)*
$\{o_1, \ldots, o_n\}$	*(one of)*	P^{-1}	*(inverse property)*
$\exists P.C$	*(exists restriction)*		

In this listing, C refers to an arbitrary concept description and P refers to a property name. We define a correspondence $\langle X, Y, Z \rangle$, where X or Y is built according to one of the rules, as *complex correspondence*. An alignment that contains a *complex correspondence* as defined as a *complex alignment*. Note also that several of these rules can be applied sequentially according to their use in standard description logics.

In the following we will talk about *matching conditions* and *correspondence patterns*. A correspondence pattern describes a special type of complex correspondence. Suppose, for example, that our algorithms detects a correspondence $\langle \exists \mathit{earlyRegistered}.\{\mathit{true}\}, \mathit{EarlyRegisteredParticipant}, \equiv \rangle$. This correspondence is a concrete instantiation of the general correspondence pattern $\langle \exists P.\{\mathit{true}\}, C, \equiv \rangle$, where P and C denote variables. A correspondence pattern can coincide with one of the construction rules listed above, but can also be compounded of several rules and might contain constants, as shown in the example.

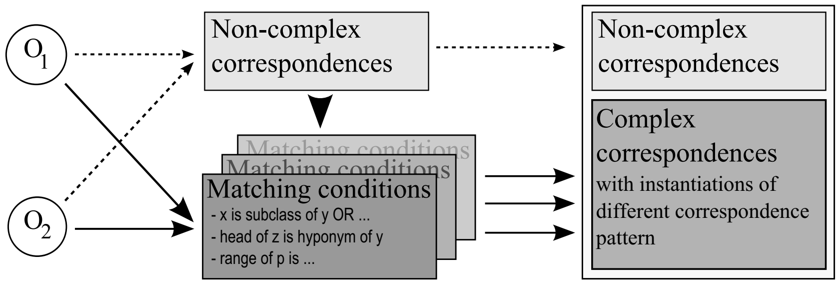


Fig. 1. Applying matching conditions to detect instances of correspondence patterns in order to generate a complex alignment.

In our approach we define for each correspondence pattern a set of matching conditions. If these conditions are fulfilled, we generate a concrete instantiation of the pattern and output a complex correspondence. The corresponding approach is depicted in Figure 1. First of all we generate by state of the art methods a non-complex alignment. This alignment is used as a kind of anchor that allows to check if certain relations hold between entities of $\mathcal{O}_1$ and $\mathcal{O}_2$ such as "is concept C in $\mathcal{O}_1$ a subconcept of concept D in $\mathcal{O}_2$". The matching conditions used to detect a certain pattern comprise structural conditions as well as the linguistic conditions are presented in Section 4.

The following section reviews related work in the field of complex ontology matching as well as recent approaches to leveraging linguistic tools and resources in ontology matching.

3 Related Work

Considering the state-of-the-art in complex ontology matching, we find recent approaches to be distinguished by three key dimensions: the design, the representation and discovery of complex correspondences. In [12], complex correspondences are mainly considered in terms of design and representational aspects. The author proposes *alignment patterns*[1] as a solution for recurring mismatches raised during the alignment of two ontologies.[2] According to Scharffe [12], complex matching is a task that has to be performed by a human user (e.g., a domain expert), who can be supported by templates for capturing complex correspondences. However, similar patterns can also be exploited by automated matching approaches, as demonstrated in this paper. The alignment patterns from [12] are expressed in terms of EDOAL[3] [5], an extension of the alignment format proposed by [3]. It covers concept and property descriptions, concept restrictions, property value transformations, comparators for restriction over entities, and variables for representing ontology entities in patterns. In this paper, we adhere to this expressive language for capturing our correspondence patterns.

Šváb-Zamazal et al. [14] consider complex ontology matching as a use case for ontology transformation. An ontology is transformed in such a way that it can be more easily matched with other ontologies. Each transformation is performed by means of a transformation pattern containing several source and target ontology patterns as well as an appropriate pattern transformation, which captures the relationships between them. Each source ontology pattern specifies detection conditions such as structural and naming conditions. The authors argue that successful non-complex matching applied to the transformed ontology can be used for finding complex correspondences by tracking changes back to the original ontology. This approach, however, lacks experimentation.

Further work related to the discovery of complex correspondences relies on machine learning techniques such as Inductive Logic Programming, for example [9]. This type of approach takes correspondences with more than two atomic terms into account, but requires the ontologies to include matchable instances – a prerequisite that is not fulfilled in many application scenarios. The approach proposed in this paper does not require the existence of instances. Moreover, we can find related work in the field of database schema matching. In [2] the authors describe complex matching as the task of

---

[1] In the ODP taxonomy of patterns the notion of an *alignment pattern* is used instead of the notion of a *correspondence pattern*. In particular, correspondence patterns are considered as more general, having alignment patterns and reengineering patterns as subcategories. However, we stick to the terminology introduced in [6] where an alignment is a set of correspondences.

[2] These patterns are now being included within *OntologyDesignPatterns.org* (ODP).

[3] http://alignapi.gforge.inria.fr/edoal.html

finding corresponding composite attributes (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 (e.g., [1]).

According to Euzenat and Shvaiko [6] *linguistic* approaches to ontology matching can be distinguished into language-based methods and methods which are based on linguistic resources, whereas the more general class of *terminological* approaches also includes string-based methods. The latter type of approach, i.e., similarity measures on the lexical layer of ontologies, is part of almost every state-of-the-art matcher. There is also a large body of work acknowledging the benefits of linguistic resources such as WordNet when it comes to detecting lexical-semantic relations between concept or property labels (see [7] for an overview). In addition, the low coverage of WordNet in certain application domains has motivated the development of methods which leverage more implicit evidence for those relationships [15], and of methods based on distributional similarities which can be computed from textual information associated with ontology entities [10, 8]. Only very few matchers, however, make use of natural language processing techniques that go beyond tokenization and lemmatization [17].This paper highlights the potential that lies within linguistic and in particular language-based methods for ontology alignment.

4 Linguistic Analysis

In order to facilitate the integration of state-of-the-art tools and resources for natural language processing into the matcher, we developed LiLA[4] (*Linguistic Label Analysis*) – a Java-based framework for the linguistic analysis of class and property labels which provides a single uniform interface to the following open-source tools:

JWNL (version 1.4.1)[5] is a programing interface for accessing the WordNet dictionary (version 3.0)[6] which contains information about more than 200,000 English words and their lexical semantic relationships.

OpenNLP (version 1.3.0)[7] is a framework for linguistic analysis including, for instance, components for determining the lexical categories of words (e.g., *adjective*).

MorphAdorner (version 1.0)[8] is a text processing framework which amongst other components provides means for morphological analysis and generation, i.e., inflection of words.

LexParser (version 1.6.3)[9] also known as the *Stanford Parser* is a syntactic parser which can be used to determine the grammatical structure of phrases (e.g., noun phrases such as “accepted paper”) or sentences.

---

[4] http://code.google.com/p/lila-project/

[5] http://sourceforge.net/projects/jwordnet/

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

[7] http://opennlp.sourceforge.net

[8] http://morphadorner.northwestern.edu

[9] http://nlp.stanford.edu/software/lex-parser.shtml

In addition, LiLA features a simple word sense disambiguation component and a spell checker. The remainder of this section illustrates the core functionalities of LiLA by virtue of a noun phrase, which serves as a running example.

paper written by clever students

Part-of-Speech Tagging. Each word in natural language belongs to a syntactic category (or *part-of-speech*), which defines its basic syntactic behavior. Accordingly, a part-of-speech tagger is a linguistic processing component for assigning appropriate syntactic categories to a given set of words. While in principle, each POS tagger may use its own set of category labels (*tags*), tag sets such as the *Penn Treebank Tag Set*[10] for English are widely used, and certain naming conventions have emerged as quasi-standards. Here, `NN` and `NNS` denote common nouns (singular or plural, respectively), `IN` stands for a preposition, `JJ` indicates an adjective and `VBN` is the tag for a past participle verb.

paper [`NN`] *written* [`VBN`] *by* [`IN`] *clever* [`JJ`] *students* [`NNS`]

Morphological Analysis. The field of morphology is concerned with the internal structure of words, more precisely the morphological rules for inflection and word-formation that enable humans to build a rich vocabulary from a basic inventory of *morphemes* – the smallest units in natural language that carry meaning. Each word consists of one or more morphemes. Words that are built from more than one morpheme can be split into a *stem* and an *affix*, i.e., a morph attached to a stem like “student”+“s”, for example. In this case, the plural “s” is an inflectional morpheme, which alters the base form (also called *lexeme*) without changing its syntactic category. A component which reduces each word to its *lemma* (i.e., the canonical form of a lexeme which is typically included in the lexicon of a language) is called a *lemmatizer*.

LiLA relies upon the MorphAdorner framework for performing both lemmatization and morphological synthesis, i.e., the generation of specific word forms (e.g., “students”) from lexemes (e.g., “student”). This also works well for irregular verbs such as “write” for which we are able to generate, for instance, the past participle (“written”) by means of *conjugation*. This way LiLA can convert between singular and plural of the same noun (*declination*), as well as between active and passive *voice* or different tenses of a given verb. The functionality to obtain derivations such as *nominalizations* of verbs (e.g., “accept” and “acceptance”), for example, is provided by JWPL.

Lexical Semantic Analysis. *Lexical semantics* is the branch of linguistics that studies the meaning of words and their relationships. The popular lexical database of WordNet, which covers a wide range of such lexical semantic relations, is queried by LiLA through the API of JWPL. Thus, given a word such as “clever” or “student”, LiLA can get access to detailed information about the possible meanings of the word (see *homonymy*), as well as about its synonyms, hyponyms, antonyms and otherwise related senses (e.g., meronyms).

Synonymy, at least *true* synonymy, is rarely found in natural language. However, there are many so-called *near-synonyms* (or *plesionyms*), i.e., words that share a common

---

[10] http://www.cis.upenn.edu/ treebank/

meaning in a given context. Hence, two words are considered synonyms (or near-synonyms) if they can be exchanged for one another in a sentence without altering its truth conditions (e.g., “student” and “scholar”).

Homonymy and polysemy are types of semantic ambiguity. Two words are considered *homonymous* if they are spelled (*homograph*) and pronounced (*homophone*) in the same way, while having distinct meanings (or *senses*). Homonyms with related meanings, are called (regular) *polysemes* (e.g., “paper” as a substance or a writing sheet made thereof). In case a query posed to the WordNet API returns multiple senses for a given concept or property label, LiLA’s **word sense disambiguation** component selects the most likely sense based on a vector-based representation of the current lexical context.[11]

Hyponymy is a kind of subordination relating one lexical unit to another one with a more general sense. The former is then called a *hyponym*, whereas the latter, i.e., the superordinate, represents the *hypernym*. Like meronymy, hyponymy is only transitive within one and the same category (e.g., functional). A verb which is more specific than another verb is sometimes called *troponym* (e.g., “write” and “create”).

Antonymy is a kind of oppositeness that mostly holds between adjectives, but also some verbs and even nouns can be considered antonyms if they exhibit opposite semantic traits. One can distinguish between different types of antonyms such as *gradable* (“early” and “late”), *complementary* (“acceptable” and “unacceptable”) and *relational* (“student” and “professor”) antonyms.

Syntactic Parsing in computational linguistics typically refers to the analysis of syntactic structures. Each parsing algorithm relies upon a certain grammar, that is a formalism developed to describe the syntactically well-formed structures in a given language. Essentially, two types of grammars – dependency grammars and phrase structure grammars – have emerged as the most wide-spread means to analyze and generate syntactically well-formed utterances. The phrase structure depicted further below (left column) has been generated by the Stanford Parser.[12] `NP` and `VP` are phrasal categories denoting a *noun phrase* or *verb phrase*, respectively.

```
(S
 (NP (NN paper))                                 nsubj(written-2, paper-1)
 (VP (VBN written)                               pobj(by-3, students-5)
   (PP (IN by)                                   amod(students-5, clever-4)
     (NP (JJ clever) (NNS students)))))          prep(written-2, by-3)
```

Given such a syntactic analysis and an appropriate set of rules for the English language, we can determine (e.g., by means of OpenNLP or the Stanford Parser) the *head* of a phrase.[13] In this case the head, i.e., the word which determines the category of a phrase and carries the most essential semantic information, is the noun “paper”. Note that the widely used heuristic of considering the right-most word as the head a phrase

---

[11] For the experiments reported in this paper, we initialized this vector by adding all of the entity labels found in the respective ontology.

[12] The phrasal category of the top-most node in the syntax tree should be `NP` (*noun phrase*) rather than `S`, which denotes a sentence.

[13] We omit the distinction between syntactic and semantic heads.

(*righthand head rule*) only works well for morphological units such as compounds (e.g., “student paper”).

In addition, the Stanford Parser provides us with a list of syntactic *dependencies* between the individual words of the phrase (right column). For example, it identifies “paper” as the *subject* of “written” and “clever” as an *adjective modifier* of “students”.

In the following, we will explain how a linguistic analysis along the dimensions outlined in this section can improve the results of a complex matching approach.

5 Matching Conditions

In the following we show how to use the linguistic analysis combined with a set of simple structural techniques to detect complex correspondences. In particular, we specify four correspondence patterns and define for each of them a set of matching conditions. If each of these conditions is fulfilled we generate a correspondence as instance of this pattern.

Class by Attribute Type (CAT) A correspondence $A_{\#1} \equiv \exists R_{\#2}.B_{\#2}$ of the *CAT* pattern is generated by our algorithm, if the following conditions hold.

1. The label of $B_{\#2}$ is the nominalization of the modifier of the label of $A_{\#1}$.
2. The class $B_{\#2}$ is subclass of the range of $R_{\#2}$.
3. One of the following two conditions holds:
   (a) The class $A_{\#1}$ is a subclass of the domain of $R_{\#2}$ due to the anchor alignment.
   (b) The label of $A_{\#1}$ is a hyponym of the label of the domain of $R_{\#2}$.

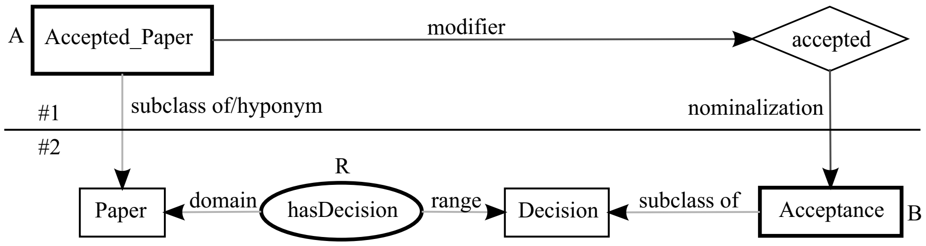


Fig. 2. Conditions relevant for detecting *CAT* correspondence $Accepted\_Paper_{\#1} \equiv \exists hasDecision_{\#2}.Acceptance_{\#2}$.

A typical example is $Accepted\_Paper_{\#1} \equiv \exists hasDecision_{\#2}.Acceptance_{\#2}$. In Figure 2 we depict the three matching conditions relevant for this example: 1) The linguistic analysis reveals that “Acceptance” is the nominalization of the active form of “Accepted”, which is in turn the modifier of “AcceptedPaper”. A morphological analysis indicates that the first condition is fulfilled. 2) We use a reasoner to check whether $Acceptance_{\#2}$ is a subclass of the range of $hasDecision_{\#2}$, and find that the second condition is fulfilled, too. 3) The third condition is a disjunction. In this concrete case the anchor alignment contains correspondence $Paper_{\#1} \equiv Paper_{\#2}$, which allows us to conclude that the third condition is fulfilled. The third condition is a disjunction, because on the one hand it might happen that the anchor alignment does not contain the required correspondence, but the linguistic analysis detects the lexical-semantic relation of hyponomy. On the other hand the linguistic analysis might fail, but the information encoded in the anchor alignment might be sufficient. We defined similar disjunctions for some of the other patterns.

In our previous approach we computed e.g., the edit-distance between “Acceptance” and “Accepted” to detect a relation between $AcceptedPaper_{\#1}$ and $Acceptance_{\#2}$. In case it exceeded a certain threshold the counterpart of the first condition was fulfilled. Similarity-based conditions are now replaced by conditions based on linguistic analysis.

Class by Inverse Attribute Type (CAT^{-1}) A correspondence $A_{\#1} \equiv \exists R^{-1}_{\#2}.\top$ of the *CAT*$^{-1}$ type is generated if the following conditions hold.

1. The label of $A_{\#1}$ is the nominalization of the active form of the label of $R_{\#2}$.
2. There exists a class $B_{\#2}$ which is a proper subclass of the range of $R_{\#2}$.
3. One of the following two conditions holds:
   (a) $A_{\#1}$ is, due to the anchor alignment, a subclass of $B_{\#2}$.
   (b) The label of $A_{\#1}$ is a hyponym of the label of $B_{\#2}$.

This pattern and the conditions to detect it are similar to the *CAT* pattern and its conditions. Due to the lack of space we omit a detailed description.

Class by Attribute Value (CAV) Here, we restrain ourselves to detect the boolean variant of the general *CAV* pattern whereby the the attribute values are *true* and *false*. Let in the following `adjm`(X) refer to the adjective modifier of the phrase X, let `advm`(X) be the adverbial modifier in X and let `vp`(X) refer to a verb phrase contained in X. A correspondence $A_{\#1} \equiv \exists R_{\#2}.\{false\}$ is generated by our algorithm, if the following conditions hold.

1. The range of the datatype property $R_{\#2}$ is `Boolean`.
2. One of the following two conditions holds:
   (a) The class $A_{\#1}$ is a subclass of the domain of $R_{\#2}$ due to the anchor alignment.
   (b) The label of $A_{\#1}$ is a hyponym of the label of the domain $R_{\#2}$.
3. `advm(label(`$R_{\#2}$`))` is the antonym of `advm(adjm(label(`$A_{\#1}$`)))`.
4. The head of `label(`$R_{\#2}$`)` is the nominalization of `vp(adjm(label(`$A_{\#1}$`)))`.

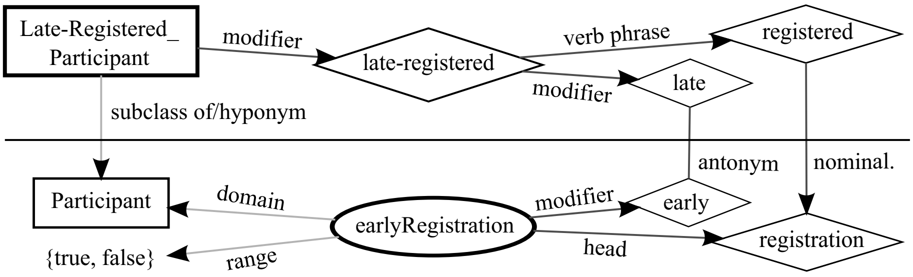


Fig. 3. Conditions relevant for detecting *CAV* correspondence $Late-Registered\_Participant_{\#1} \equiv \exists earlyRegistration_{\#2}.\{false\}$.

Regarding this pattern we use the linguistic analysis to detect antonyms. We expect that modifiers, which are antonyms, will be used to describe a pair of disjoint classes. Complex expressions as $\exists R_{\#2}.\{true\}$ and $\exists R_{\#2}.\{false\}$, given that $R_{\#2}$ is a functional property, refer also – for logical reasons – to a pair of disjoint classes.

Inverse Property (IP) A correspondence $R^{-1}_{\#1} \subseteq P_{\#2}$ of type *IP* is generated, if all following conditions hold.

1. The verb phrase of the label of $R_{\#1}$ is the active voice of the verb phrase of the label of $P_{\#2}$.
2. One of the following two conditions holds:
   (a) The domain of $R_{\#1}$ is a subclass of the range of $P_{\#2}$.
   (b) The label of the domain of $R_{\#1}$ is a hyponym of the label of the range of $P_{\#2}$.
3. One of the following two conditions holds:
   (a) The range of $R_{\#1}$ is a subclass of the domain of $P_{\#2}$.
   (b) The label of the range of $R_{\#1}$ is a hyponym of the label of the domain of $P_{\#2}$.

The IP pattern is the simplest pattern regarding its set of conditions. The first condition is based on the fact that two properties are inverse properties with higher probability, if both contain the same verb phrase in a different voice (active or passive voice). The two other structural conditions ensure that there is a subsumption (possibly equivalence) relation between domain and range of both properties.

It is surprising that it is sometimes harder to detect a simple property equivalence or subsumption than an instance of the *IP* pattern. An example found in our experiments is the following one. In one ontology we have the property $writtenBy_{\#1}$ and its inverse $authorOf_{\#1}$, while in the other ontology we have a property $write\_paper_{\#2}$. Regarding these properties there are two correct correspondences, namely $authorOf_{\#1} \subseteq write\_paper_{\#2}$ and $writtenBy^{-1}_{\#1} \subseteq write\_paper_{\#2}$. While the first one is hard to detect, the second one fulfills all of the conditions listed above and is thus detected by our approach. It is now possible to derive the first correspondence from the second one.

6 Experiments

For our experiments we used the dataset of the OAEI conference track [16]. This dataset consists of several, relatively expressive ontologies that describe the domain of organizing conferences from different perspectives. Regarding this dataset, we made experiments on the full set of ontologies.[14] However, since we want to compare our approach to its predecessor, we also present results restricted to a subset of 9 ontologies that we have used in [11]. To decide whether the structural conditions hold, we used the Pellet reasoner [13]. For checking the linguistic conditions we rely on LiLA. Structural conditions that compare entities from different ontologies require an anchor alignment of non-complex correspondences. We generated this alignment by computing a similarity measure based on the Levensthein distance thresholding the results at a high value.

---

[14] Since we could not process the ontologies LINKLINGS, COCUS, CONFIOUS with the Pellet reasoner we have excluded them from our experiments.

Dataset	Approach	True Positives					False Positives					Precision
		CAT	CAT^{-1}	*CAV*	*IP*	$\sum$	*CAT*	CAT^{-1}	*CAV*	*IP*	$\sum$	
subset	similarity	4	3	2	-	9	4	7	0	-	11	0.450
subset	linguistic	4	2	2	8	16	1	0	0	0	1	0.941
full set	linguistic	9	4	2	17	32	5	1	0	1	7	0.821

Table 1. Experimental results in terms of true and false positives.

In Table 1, the approach described in this paper is referred to as *linguistic* approach, its predecessor is called *similarity* approach. For each of the four patterns we show the number of true and false positives, as well as the precision of the approach. We can see that the use of linguistic methods helped us to increase precision of the overall approach by a large degree from 45% to 94%, while the number of correctly detected correspondences stays nearly stable. Notice that based on the similarity approach it was not possible to define matching conditions for the *IP* pattern. If we include the IP pattern in our analysis, we come up with the conclusion that we signficantly increased recall.

With the similarity approach we could additionally define conditions for a pattern called *property chain* (not depicted here). We omitted this pattern here, as we thought that the rationales underlying this pattern are hard to justify and that the chosen conditions were partially overfitting to certain aspects of the dataset. Note that the values given for the similarity approach are based on the optimal threshold: raising or lowering the threshold results in a clear loss of recall or precision, while the additional gain is rather limited as shown in [11]. The linguistic methods are different in that there is no threshold whose value would be crucial for the overall performance of the approach.

Our previous approach has been criticized for the requirement of a correct and complete input alignment. However, the new results indicate that such an input alignment is not necessary. It is sufficient to generate an incomplete and partially incorrect alignment in a prior step and to use it as an anchor in the subsequent matching process. Thus, the approach is robust against the noise introduced by such an imperfect anchor alignment.

7 Conclusion

In this paper we have described how to integrate linguistic techniques into a pattern-based approach for detecting complex correspondences. In particular, we have presented correspondence patterns and defined for each of them a set of matching conditions. While in a previous approach [11] many of these conditions were based on computing a simple string-based similarity value, we argued now that it is more appropriate to substitute these conditions by a set of conditions that make use of a linguistic analysis. In our experiments we showed that the new approach yields a significantly higher precision. The tool used to conduct the experiments is open source and available online.[15] Due to its modular structure, matching conditions for new correspondence pattern can easily be specified in a generic XML syntax.

Acknowledgments Johanna Völker is financed by a Margarete-von-Wrangell scholarship of the European Social Fund (ESF) and the Ministry of Science, Research and the Arts Baden-Württemberg. Ondřej Šváb-Zamazal is partly supported by grant no. P202/10/1825 of the Grant Agency of the Czech Republic.

---

[15] http://code.google.com/p/generatingcomplexalignments/

References

1. Robin Dhamankar, Yoonkyong Lee, Anhai Doan, Alon Halevy, and Pedro Domingos. iMAP: discovering complex semantic matches between database schemas. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2004.
2. AnHai Doan and Alon Y. Halevy. Semantic-integration research in the database community. *AI Magazine*, 26:83–94, 2005.
3. Jérôme Euzenat. An API for ontology alignment. In *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, pages 698–712, 2004.
4. Jérôme Euzenat, Alfio Ferrara, Laura Hollink, Antoine Isaac, Cliff Joslyn, Véronique Malaisé, Christian Meilicke, Andriy Nikolov, Juan Pane, Marta Sabou, François Scharffe, Pavel Shvaiko, Vassilis Spiliopoulos, Heiner Stuckenschmidt, Ondřej Šváb-Zamazal, Vojtěch Svátek, Cássia Trojahn dos Santos, George A. Vouros, and Shenghui Wang. Results of the ontology alignment evaluation initiative 2009. In *Proceedings of the 4th International Workshop on Ontology Matching (OM-2009)*, 2009.
5. Jérôme Euzenat, François Scharffe, and Antoine Zimmermann. Expressive alignment language and implementation. Deliverable 2.2.10, Knowledge web, 2007.
6. Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, 2007.
7. Feiyu Lin and Kurt Sandkuhl. A survey of exploiting WordNet in ontology matching. In Max Bramer, editor, *IFIP AI*, volume 276 of *IFIP*, pages 341–350. Springer, September 2008.
8. Giuseppe Pirrò and Domenico Talia. LOM: a linguistic ontology matcher based on information retrieval. *Journal on Information Science*, 34(6):845–860, 2008.
9. Han Qin, Dejing Dou, and Paea 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.
10. Yuzhong Qu, Wei Hu, and Gong Cheng. Constructing virtual documents for ontology matching. In *Proceedings of the 15th International Conference on World Wide Web (WWW)*, pages 23–31, New York, NY, USA, 2006. ACM.
11. Dominique Ritze, Christian Meilicke, Ondřej Šváb-Zamazal, and Heiner Stuckenschmidt. A pattern-based ontology matching approach for detecting complex correspondences. In *Proceedings of the ISWC workshop on ontology matching*, Washington DC, USA, 2009.
12. François Scharffe. *Correspondence Patterns Representation*. PhD thesis, University of Innsbruck, 2009.
13. Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: a practical OWL-DL reasoner,. *Journal of Web Semantics*, 5:51–53, 2007.
14. Ondřej Šváb-Zamazal, Vojtěch Svátek, and Luigi Iannone. Pattern-based ontology transformation service exploiting OPPL and OWL-API. In *Knowledge Engineering and Knowledge Management by the Masses. EKAW-2010.*, 2010. Accepted.
15. Willem Robert van Hage, Sophia Katrenko, and Guus Schreiber. A method to combine linguistic ontology-mapping techniques. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *International Semantic Web Conference (ISWC)*, volume 3729 of *LNCS*, pages 732–744. Springer, November 2005.
16. Ondřej Šváb, Vojtěch Svátek, Petr Berka, Dušan Rak, and Petr Tomášek. Ontofarm: Towards an experimental collection of parallel ontologies. In *Poster Track of the International Semantic Web Conference (ISWC)*, Galway, Ireland, 2005.
17. Pinar Wennerberg, Manuel Möller, and Sonja Zillner. A linguistic approach to aligning representations of human anatomy and radiology. In *Proceedings of the International Conference on Biomedical Ontologies (ICBO)*, 7 2009.

Lost in Translation? Empirical Analysis of Mapping Compositions for Large Ontologies

Anna Tordai,[1] Amir Ghazvinian,[2] Jacco van Ossenbruggen,[1] Mark A. Musen,[2]
Natalya F. Noy[2]

[1] VU University Amsterdam, Amsterdam, The Netherlands
{atordai,jrvosse}@few.vu.nl
[2] Stanford University, Stanford, CA 94305, US
{amirg,musen,noy}@stanford.edu

Abstract. When three or more ontologies have been aligned, longer chains of mapped concepts start to appear. In this paper, we empirically study the nature of these composite mappings, focusing on chains of (near) equivalence links of length two. We ask human experts to evaluate samples of composite mappings, taken from large real life data sets. Based on these evaluations, we analyze the features of mappings produced by composition in three different domains (bio-medicine, cultural heritage, and library subject headings), among ontologies in multiple languages (English, Dutch, German, and French), and using existing mappings that were created by different methods (lexical and instance-based methods). We examine the quality of the composite mappings relative to the quality of the input mappings and analyze how characteristics of the input mappings and the ontologies influence the composition.

1 Introduction

Researchers typically study ontology alignments in the context of a single source and target ontology. As more and more of such alignments are being created and published, however, longer chains of equivalent or otherwise related concepts start to emerge in our data sets. In this paper, we analyze the quality of a subset of such chains, focusing on short chains of equivalence and near equivalence links. Most of us have clear intuitions about the properties of such chains. For example, equivalence relations such as `owl:sameAs` and `skos:exactMatch`, are defined as being transitive, so it should be safe to assume that if term A is equivalent to B, and B is equivalent to C, then A should also be equivalent to C. We will test this hypothesis empirically by determining to what extent such transitivity actually holds in our data sets, and if not, what is going wrong Furthermore, for relations such as `skos:closeMatch`, which are not defined as being transitive, we might ask how often chains of these relations turn out to be transitive after all.

We use the notion of a mapping as defined in [15]. Given a mapping from A to B and from B to C, where concepts A, B and C are part of three different ontologies, we call the mapping from A to C a *composite mapping*. Although mapping composition is related to the use as background knowledge where concept B would be part of the background ontology [2], we do not predefine ontologies as a source of background knowledge. We analyze the properties of such composite mappings on real life data sets, addressing the following two research questions:

- What is the quality of composite mappings relative to the quality of input mappings?
- Does the quality of composite mappings depend on other characteristics of the input mappings or ontologies?

In order to answer these research questions, we study composite mappings for ontologies in different domains, using input mappings generated in different ways (Section 3.1). We analyzed the precision of composite mappings by sampling them and having human experts verify the samples (Section 3.3). In some cases, we already had pre-existing alignments for the sets of ontologies for which we analyze composite mappings. In these cases, we compared the precision of the composed mappings with the precision of existing mappings. We then analyzed our results (Section 5) and made observations regarding the quality and quantity of composed mappings, trying to identify reasons for correct and incorrect mapping compositions based on characteristics of the data and the input mappings.

The main contribution of this paper is a large-scale empirical analysis of the nature of composite mappings given varied sets of input ontologies and mappings.

2 Related Work

Researchers in the area of database schema matching have studied mapping composition extensively [1, 14, 4]. However, these researchers have focused on the use of mapping composition to perform query transformation rather than for generating mappings.

In ontology matching, Euzenat [6] discusses mapping composition in a theoretical paper on algebras of relations as a means for validating existing mappings and creating new mappings. This work considers composition through equivalence mappings to be a trivial case because the result is an equivalence relation, and because we can assume that equivalence is transitive. In practice, however, automatically generated mappings are usually similarity mappings at best, and therefore the composition of such mappings is not trivial. We look at such automatically generated mappings and analyze results of composition to find out whether they are interesting or truly lost in translation.

Researchers have already developed a plethora of tools for generating mappings and compared their performance at the OAEI. These off-the-shelf tools, such as ASMOV [12], RiMOM [22], Falcon-AO [11], and DSSim [16] perform well on OAEI benchmarks and on certain specialized tracks. However, the results of the 2009 library track showed that current tools largely fail on extremely large vocabularies and vocabularies that use multiple languages [7].

Mapping composition has some parallels to the use of background knowledge by mapping tools. Tools such as SAMBO [13] and ASMOV use background knowledge (UMLS Metathesaurus, WordNet) to improve the quality of mappings. When mapping two domain ontologies, these tools either use existing mappings from these domain ontologies to some background source, such as UMLS or WordNet, or create these mappings "on the fly" through lexical comparison or other means. The tools then use

Set	Domain	Ontologies	Language	Ontology size
BioPortal	Biomedicine	151 ontologies from BioPortal	English	Ranging from under 100 concepts to 380K concepts Mean size=17,805 (SD= 61,614) Total concepts: 2,688,609
CH	Cultural Heritage	Getty's Art and Architecture Thesaurus (AAT)	English and Dutch	27,077 concepts with English and Dutch labels
		Princeton WordNet	English	115,424 synsets with 203,147 English labels
		Cornetto	Dutch	70,370 synsets and 103,762 Dutch labels
Library	General	Library of Congress Subject Headings (LCSH)	English	339,612 concepts
		Rameau	French	157,287 concepts
		SWD	German	163,175 concepts

Table 1. Sets of ontologies that we used in mapping composition and their characteristics.

these mappings to a single source of background knowledge for creating mappings for the domain ontologies. This method is related to mapping composition because we use a mapping to a third ontology or vocabulary. In this sense, in mapping composition *any* ontology becomes a source of background knowledge.

The COMA [5] and COMA++ [3] tools combine several matching techniques including composition of mappings. The evaluation of the tools demonstrated the effectiveness of mapping composition without going into a more detailed analysis of the results.

3 Materials and Methods

In this section, we describe the ontologies and existing mappings that we used for mapping composition (Section 3.1), the method for creating compositions and its complexity (Section 3.2), and our methodology for assessing the precision of the composed mappings (Section 3.3).

3.1 Data: Ontologies and Input Mappings

In order to get a comprehensive analysis of mapping composition under different conditions, we considered three sets of ontologies and mappings. We have ontologies in three different domains: biomedicine, cultural heritage and library subject headings (Table 1). The terms in these ontologies have labels in four languages: English, Dutch, German and French, and the input mappings we use for composition were generated using two types of methods: lexical method, and instance-based method (Table 2).

Set	Method for mapping generation	Number of mappings used for composition	Precision
BioPortal	Lexical comparison of preferred names and synonyms	575,642 mappings	0.94
		459,941 Preferred–Preferred	0.99
		115,701 Preferred–Synonym	0.76
CH	Lexical comparison of labels	6,914 AAT–Cornetto	0.88
		4,592 AAT–WordNet	0.82
		3,144 Cornetto-WordNet	0.95
Library	Instance-based	2,242 LCSH–Rameau	0.95
		2,334 SWD–LCSH	0.54
		685 Rameau–SWD	0.72

Table 2. Input mappings that we used for mapping composition for the three sets of ontologies.

Our first set of ontologies came from BioPortal [17], a Web-based repository of biomedical ontologies. At the time we collected the data, BioPortal contained 151 ontologies with more than 2.5 million concepts among them. We generated mappings between these ontologies using simple lexical comparisons of preferred names and synonyms after normalization [9, 8].

The second set of mappings links three large vocabularies in the cultural-heritage domain: Getty's Art and Architecture Thesaurus (AAT[3], extended with Dutch labels from AATNed[4]), Princeton WordNet[5] version 2.0 and Cornetto,[6] a WordNet-like lexical resource for Dutch. We generated mappings between AAT and WordNet, and between AAT and Cornetto using simple lexical comparison [19]. The Cornetto project [20] created mappings between Cornetto and different versions of WordNet using a combination of manual and automatic methods.

Finally, we used a set of ontologies and mappings from the Library track in the OAEI 2009. This set contains three lists of subject headings for describing content of books: the Library of Congress Subject Headings (LCSH); Rameau, a list used by the French National Library; and the Subject Heading Authority File (SWD), which is used by the German National Library. Each list contains from 150,000 to 300,000 concepts.

We used the mappings that Wang and colleagues [21] created using instance-based matching based on books that were classified using terms from more than one vocabulary. This method for generating mappings ranks the resulting mappings according to confidence level. Although there are a total of almost 2 million mappings, over 90% of them have confidence measure lower than 0.1. For the purpose of composing mappings, we selected only those mappings that had a confidence measure greater than 0.7.

---

[3] `http://www.getty.edu/research/conducting_research/vocabularies/aat/`

[4] `http://www.aat-ned.nl/`

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

[6] `http://www2.let.vu.nl/oz/cornetto/index.html`

We estimate the precision of these mappings by evaluating samples manually. These mappings involve fewer than 1.5% of the concepts in the vocabularies.

In the cultural heritage and OAEI library track the number of input mappings is significantly lower than in the BioPortal case, as our aim was to select high-quality mappings. We chose a representative subset in order to analyze the properties of mapping composition.

3.2 Computing Mapping Composition

In this paper, we consider only composition of two mappings. The BioPortal compositions were computed using a relational database, and the cultural heritage and OAEI library track composition algorithms were written in SWI-Prolog.[7].

Since we had detailed information on the source of the input mappings for all BioPortal ontologies, we further analyzed the composed mappings for BioPortal to understand better how characteristics of input mappings affect the number and precision of composed mappings. To perform such analysis, we broke down the composed mappings into groups, based on the types of input mappings that contributed to the composition. There are two types of input mappings (see Table 2): Preferred–Preferred and Preferred–Synonym mappings. We do not include Synonym–Synonym mappings in our input because they have low precision(0.36). Different combinations of the input mappings produce six groups of composed mappings which are displayed in Figure 1.

For instance, Figure 1A illustrates the case where we composes a mapping from a preferred name for the concept C_1 to a preferred name for C_2 with a mapping from the preferred name for C_2 to the preferred name of C_3. We refer to this case as PPP. Note that this composition produces a subset of the Preferred–Preferred mappings between O_1 and O_3. PSP mappings (Figure 1B) also produce a subset of the Preferred–Preferred mappings. Similarly, PPS mappings (Figure 1C) and SPS mappings (Figure 1D) produce subsets of the Preferred–Synonym and Synonym–Synonym mappings between O_1 and O_3, respectively. We analyze these subsets and compare their precisions to those of the original Preferred–Synonym and Synonym–Synonym mappings that were generated directly by comparing O_1 and O_3. Figure 1E and F illustrate the other two cases, $PSPS$ and $PSSP$, which produce mappings that we cannot obtain by comparing preferred names and synonyms directly.

3.3 Sampling and evaluation

In order to evaluate the precision of the composed mappings as well as the precision of input mappings (see Table 2), we sampled the mappings and evaluated the samples manually. Because of the scale of our data—with hundreds of thousands of mappings to verify—evaluating all the mappings manually was not feasible. Furthermore, because of the size of the ontologies themselves, creating a complete set of mappings so that we can evaluate recall was not feasible either. In addition, the recall of mapping composition is necessarily limited by the recall of the input mappings used for composition. Thus, we focus in this evaluation on estimating only the precision of the composed mappings.

---

[7] `http://www.swi-prolog.org/`

For BioPortal mappings, we used stratified sampling [10] to select mappings for manual evaluation. Among the BioPortal ontologies, there is a large number of ontology pairs that have only one or two composed mappings between them. At the same time, there are pairs of ontologies that have thousands of mappings between them. Therefore, we constructed the strata to ensure that our samples include mappings between ontology pairs with only a few mappings between them, as well as mappings between ontology pairs with thousands of mappings, and clusters in between. We sampled a total of 2350 mappings from the different BioPortal mappings sets. Our sample sizes ranged from 210 to 400 mappings per set depending on the number of original mappings.

In the case studies involving cultural heritage and library subject headings, we evaluated manually all mapping sets containing fewer than 500 mappings and took samples of 100 mappings from larger sets. We sampled the total of approximately 1,000 mappings from these sets.

Human experts evaluated the samples using the evaluation tool used in [18] for the cultural heritage and Library track data, and a similar tool for the BioPortal data, and categorized each mapping into one of six categories: *exact match*, *close match*, *broader match*, *narrower match*, *related match*, or *incorrect*. For measuring precision, we considered only exact and close matches as correct. A detailed analysis of the broader, narrower and related matches is out of scope for this paper but we plan to perform it in the future. We measured agreement using Cohen's kappa on subsets of samples between raters, finding substantial agreement for BioPortal (0.72) and cultural heritage evaluation (0.70) and almost perfect agreement with the manually evaluated mappings used in the OAEI library track (0.85).

4 Results

In this section, we present the precision of mapping composition for the three sets of ontologies in our study. We discuss these results in Section 5.

4.1 Results: Biomedical Ontologies

Figure 2A shows the results for the overall precision of composed mappings. Using 575,642 input mappings with precision 0.94, we generated 599,625 composed map-

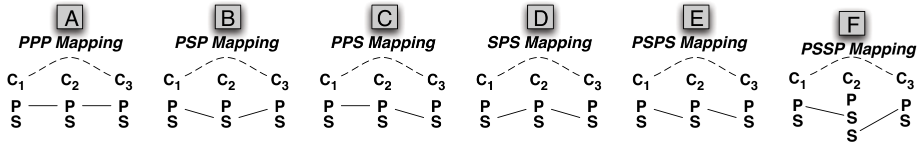


Fig. 1. Methods for composing mappings between concepts in three different ontologies ($C_1 \in O_1$, $C_2 \in O_2$, $C_3 \in O_3$) using mappings between preferred labels (P) and synonyms (S). Figure A illustrates the PPP mappings: a composition of a mapping from a preferred name of C_1 to preferred name of C_2 with the mapping between preferred names of C_2 and C_3. Figure B illustrates PSP mappings: a match of C_1 preferred name to C_2 synonym with a match of C_2 synonym to C_3 preferred name. Figures C-F illustrate the remaining possible cases.

pings with a precision of 0.92. Figures 2B, 2C, and 2D show the precision of composition for different cases from Figure 1. We group these cases by the sets of input mappings that they used. Composing Preferred–Synonym mappings, which had a precision of 0.76, yielded 147,438 composed mappings with precision 0.84. Other combinations (Figures 2C and 2D) resulted in sets of composed mappings with precisions similar to the precisions of the input mappings.

Figure 3 provides additional information on the precision of the individual cases. The two cases that resulted in the subset of what we could have obtained directly by comparing preferred names lexically (PPP and PSP), provided mappings with the highest precision, 0.99. The SPS mappings constitute a subset of the Synonym–Synonym mappings for O_1 and O_3. We did not use these types of mappings as input mappings because they have very low precision, 0.36. However, using mapping composition to identify a subset of Synonym–Synonym mappings almost doubles the precision of these mappings, bringing it up to 0.6.

Additionally, using composition, we identified mappings without lexical similarity in their preferred names or synonyms ($PSPS$ and $PSSP$ mappings). Such mappings can be identified by composition through a concept with lexical similarity to both mapped concepts. These two cases produced 50,353 new mappings with the precision of 0.68. For example, we found a $PSSP$ mapping between the concept CRANIAL_SKELETON from the Amphibian gross anatomy ontology and SKULL from the Foundational Model of Anatomy. These two concepts each map to the concept CRANIUM from the Teleost anatomy and development ontology, which has the synonyms CRANIAL SKELETON and SKULL.

4.2 Results: Cultural Heritage

Figure 4A shows the results of mapping composition for the cultural heritage domain. The precision of composed mappings is at least 0.8 in all three cases, with the number of mappings identified through composition ranging from 263 to 1,774. In fact, the composed mappings between Cornetto and WordNet have a precision of 0.9.

Because we have lexical mappings available for this set, we can compare the composed mappings to the lexical ones, and analyze how many non-lexical mappings we generate by composing lexical mappings.

Upon closer examination of the mappings, we found that 134 (30%) of the composed mappings between AAT and WordNet have little or no lexical similarity. For example, through composition we mapped TOBACONNISTS' SHOP to TOBACCO SHOP and WATCHMEN to GUARD. Similarly, we found 110 non-lexical mappings between AAT and Cornetto, such as BADKLEDING to BADKOSTUUM, both of which mean "bathing suit" in Dutch. This subset of composed mappings not including lexical similarity has a precision of 0.56, which is lower than the precision of composed mappings in general.

Between Cornetto and WordNet, 1,208 of the 1,774 composed mappings are listed as "near equal synonym" mappings in the original Cornetto-WordNet mappings of the Cornetto project. These are not the same as the "equal synonym" mappings used as input mappings for other compositions. Another 448 of the composed mappings are entirely new and have an average precision of 0.7.

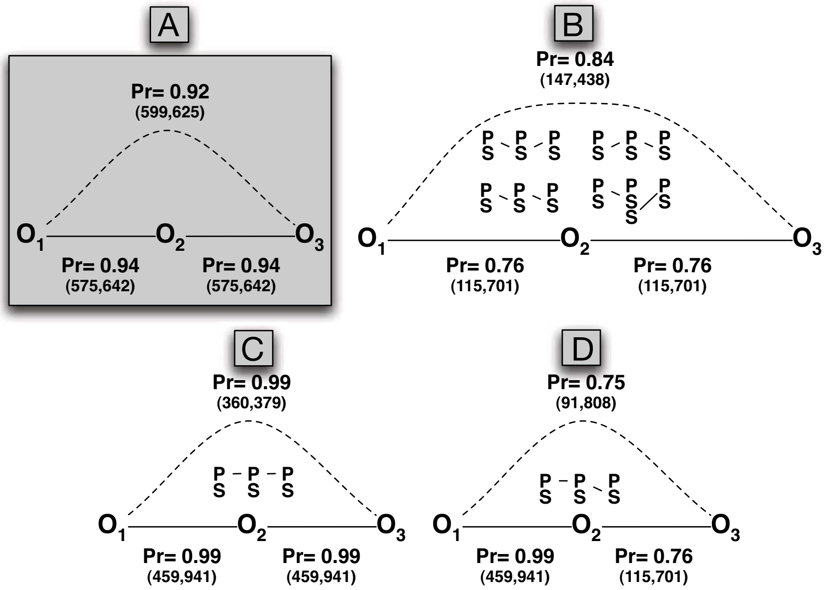


Fig. 2. Mapping composition results for BioPortal ontologies. O_1, O_2 and O_3 represent any three ontologies linked through mappings from Bioportal. Figure A (the shaded diagram) shows the overall precision of the input mappings and their number in parentheses on the solid lines. It shows the precision of composed mappings and their number above the dotted line. Figures B, C, and D provide details for the precision of composed mappings, grouped by the precision of input mappings. Figure B contains the mappings that used only Preferred–Synonym mappings as input; Figure C contains the mappings that composed Preferred–Preferred mappings; and Figure D provides the data for the composition of Preferred–Preferred mappings and Preferred–Synonym mappings.

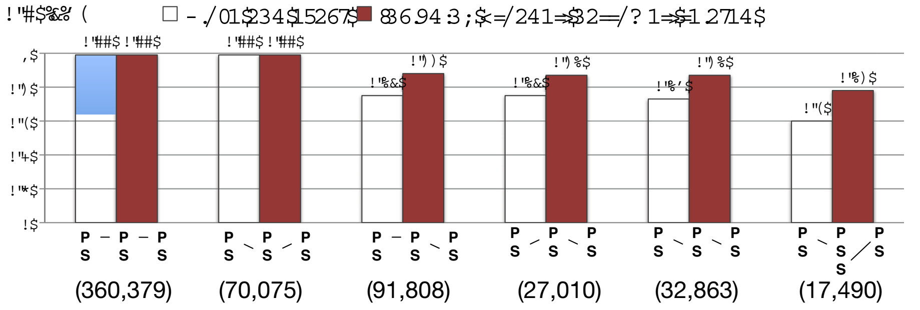


Fig. 3. Mapping composition results for BioPortal ontologies. The bar graph shows precision for composed mappings. The (lighter) left bar shows precision of exact and close matches, the (darker) right bar shows the precision if we include broader, narrower, and related matches. Numbers in parentheses indicate the total number of mappings

4.3 Results: The OAEI Library Track

Figure 4B shows the results of mapping composition using the library subject headings mappings. Precision of the composed mappings is higher than 0.74 and the number

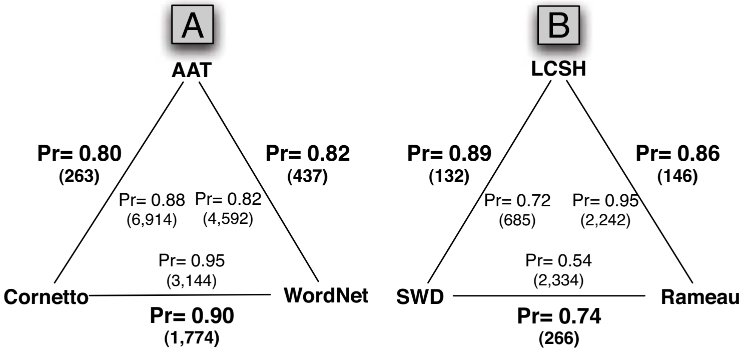


Fig. 4. Mapping composition results for cultural heritage domain (Figure A) and OAEI library track (Figure B). The numbers in bold outside the triangle show the precision and the number of composed mappings in parentheses. The numbers inside the triangle show the precision and number of input mappings

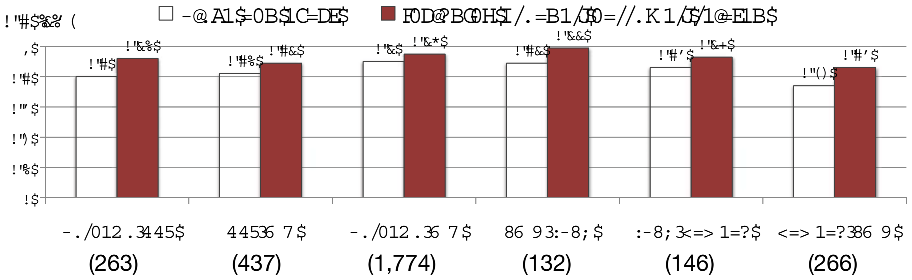


Fig. 5. Mapping composition results for cultural heritage and library-track ontologies. The bar graph shows precision for composed mappings. The (lighter) left bar shows precision of exact and close matches, the (darker) right bar shows the precision if we include broader, narrower, and related matches. Numbers in parentheses indicate the total number of mappings

of generated mappings ranges from 132 between the Subject Heading Authority File (SWD) and the Library of Congress Subject Headings (LCSH) and 266 between SWD and Rameau (a list used by the French National Library).

In two cases—mappings between SWD and LCSH and mappings between Rameau and SWD—the composed mappings actually had higher precision than the input mappings.

In this case, we also compared the composed mappings to the input mappings. We found that, of the 132 mappings between SWD and LCSH, 13 (10%) mappings did not overlap with any of the original instance-based mappings, including those that had a confidence measure lower than 0.7. In other words, for these 13 mappings, there were no instances (books) available. For LCSH and Rameau, we found 8 (5%) such “new” mappings, and for Rameau and SWD, 65 (24%) mappings. The high number of new composed mappings between Rameau and SWD is due to the low number of instances available for creating the original mappings. However, the precision of these subsets is lower: 0.37 between LCSH and Rameau, 0.54 between Rameau and SWD, and 0.92 between SWD and LCSH.

4.4 Broader, Narrower, and Related Mappings

When evaluating the composed mappings, we have also recorded whether each mapping represented a narrower, broader, or related mapping, rather than a close or exact match. Figures 3 and 5 show the increase in precision of composed mappings if we also count broader, narrower, and related mappings as correct. Figure 5 includes the data for both the cultural-heritage and the library-track case. The increase in precision in both of these cases is less dramatic than for the biomedical ontologies. In this case, the average increase in precision was 11%, whereas for BioPortal ontologies the average increase was 14%, with the most significant increase (30%) in the $PSSP$ case.

5 Discussion

In this paper, we have presented the results of our analysis of mapping composition in three different domains. Our results show that the quality of composed mappings depend on the ontology characteristics, and the content and quality of the input mappings.

The characteristics of the ontologies, such as the way they are implemented or the way preferred labels and synonyms are used, have a profound effect on composition. For example, in Medical Subject Headings (MeSH) concepts often have narrower terms as synonyms. The concept TREMORS in MeSH has a synonym NERVE TREMOR, which in reality is a narrower term, not a synonym. As a result, many of the composed mappings that involved MeSH terms were not close matches but rather broader or narrower mappings.

It is clear that the number of input mappings determines the number of composed mappings, but we see in our results that there are large variations in the number of input mappings in three cases studies. This is partly due to the size of the ontologies and partly because of the confidence level of the original input mappings which is a limiting factor for example in the Library track case study.

The content of the ontologies also influences the quality of the mapping compositions. When the content overlaps, meaning the domains of the ontologies are the same or very similar, the meaning of the concepts is also closer, and the composed mappings are likely to be equivalence mappings rather than broader, narrower or related mappings. In the cultural heritage case study Cornetto and WordNet are unlikely to cover art and architectural concepts, reducing the chance of creating equivalence compositions between AAT and Cornetto, and AAT between AAT and WordNet.

Finally, the quality of input mappings has a direct effect on the quality of mapping compositions. High quality input mappings tend to result in high quality mapping compositions. Intuitively, the precision of the compositions should be the product of the precisions of the input mappings. However, especially in the BioPortal data we find cases where the precision of compositions exceeds the precision of input mappings (Figure 3). We find similar cases in both the AAT and Library track case studies.

We also found that many of the composed mappings though not exact, or close matches nevertheless represent a semantic relationship such as broader, narrower or related (Figure 3 and 5). For example, the concept BLURRED VISION from the “Suggested Ontology for Pharmacogenomics” maps to composition to VISION ABNORMAL in the “WHO Adverse Reaction Terminology”, forming a narrower relationship between the two concepts. This kind of semantic drift between concepts seems to arise often through mapping composition caused by ontology characteristics, or concepts deviating in meaning in different languages.

In our future work we plan to perform a more detailed evaluation of the content of the mappings to determine why the precision of the composed mappings exceeds the precision of the input mappings in certain cases. We also need to study the effect of semantic drift by analyzing the relationship between the type of equivalence relation in input mappings and compositions, and extend our scope to longer composition chains.

6 Conclusion

In this paper, we presented an empirical analysis of the quality of mapping composition for various use cases. We conclude that mapping composition produced mappings of comparable quality to the input mappings; precision of the composed mappings is not much worse than the original precision of mappings, and sometimes it is even better. Even when composing lexical mappings, in some cases we produced mappings that lexical methods would not produce. Finally, the quality and the number of composed mappings can be affected significantly by the characteristics of the ontologies themselves, the type of input mappings, the number and coverage of the input mappings. Our results confirm our intuitions on mapping composition. The contribution of this paper is that we have tested these intuitions empirically and validated them using methods described in literature.

Acknowledgments

Many thanks to Guus Schreiber and Bob Wielinga for their help and comments on this paper. We thank Antoine Isaac, Shenghui Wang and the Telplus project for providing the library track data and their mappings, and for the many explanations they provided. We also thank the Cornetto project for providing data and mappings and Babette Heyer for helping with the evaluation of the mappings. This work was supported in part by the National Center for Biomedical Ontology, under roadmap-initiative grant U54 HG004028 from the National Institutes of Health, and partly by the EC eContentplus programme via de EuropeanaConnect project.

References

1. K. Aberer, P. Cudré-Mauroux, and M. Hauswirth. A framework for semantic gossiping. *SIGMOD Rec.*, 31(4):48–53, 2002.
2. Z. Aleksovski, M. C. A. Klein, W. ten Kate, and F. van Harmelen. Matching unstructured vocabularies using a background ontology. In *EKAW*, pages 182–197, 2006.
3. D. Aumueller, H.-H. Do, S. Massmann, and E. Rahm. Schema and ontology matching with coma++. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 906–908. ACM, 2005.
4. P. A. Bernstein, T. J. Green, S. Melnik, and A. Nash. Implementing mapping composition. In *VLDB '06: Proceedings of the 32nd international conference on Very large data bases*, pages 55–66. VLDB Endowment, 2006.
5. H.-H. Do and E. Rahm. Coma: a system for flexible combination of schema matching approaches. In *VLDB '02: Proceedings of the 28th international conference on Very Large Data Bases*, pages 610–621. VLDB Endowment, 2002.

6. J. Euzenat. Algebras of ontology alignment relations. In A. P. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. W. Finin, and K. Thirunarayan, editors, *International Semantic Web Conference*, volume 5318 of *Lecture Notes in Computer Science*, pages 387–402. Springer, 2008.
7. J. Euzenat, A. Ferrara, L. Hollink, A. Isaac, C. Joslyn, V. Malaisé, C. Meilicke, A. Nikolov, J. Pane, M. Sabou, F. Scharffe, P. Shvaiko, V. Spiliopoulos, H. Stuckenschmidt, O. Sváb-Zamazal, V. Svátek, C. T. dos Santos, G. A. Vouros, and S. Wang. Results of the ontology alignment evaluation initiative 2009. In *OM*, 2009.
8. A. Ghazvinian, N. F. Noy, C. Jonquet, N. Shah, and M. A. Musen. What four million mappings can tell you about two hundred ontologies. In *ISWC '09: Proceedings of the 8th International Semantic Web Conference*, pages 229–242. Springer-Verlag, 2009.
9. A. Ghazvinian, N. F. Noy, and M. A. Musen. Creating mappings for ontologies in biomedicine: Simple methods work. In *AMIA Annual Symposium (AMIA 2009)*, San Francisco, CA, 2009.
10. W. V. Hage, A. Isaac, and Z. Aleksovski. Sample evaluation of ontology-matching systems. *Proceedings of the ISWC workshop on Evaluation of Ontologies and Ontology-based tools*, pages 41–50, 2007.
11. W. Hu and Y. Qu. Falcon-AO: A practical ontology matching system. *Web Semant.*, 6(3):237–239, 2008.
12. Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka. ASMOV: Results for OAEI 2009. In *OM*, 2009.
13. P. Lambrix, H. Tan, and Q. L. 0002. Sambo and sambodtf results for the ontology alignment evaluation initiative 2008. In *OM*, 2008.
14. J. Madhavan and A. Y. Halevy. Composing mappings among data sources. In *VLDB '2003: Proceedings of the 29th international conference on Very large data bases*, pages 572–583. VLDB Endowment, 2003.
15. A. Miles and S. Bechhofer. Skos simple knowledge organization system reference, August 2009.
16. M. Nagy, M. Vargas-Vera, and P. Stolarski. Dssim results for oaei 2009. In *OM*, 2009.
17. N. F. Noy, N. H. Shah, P. L. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D. L. Rubin, M.-A. D. Storey, C. G. Chute, and M. A. Musen. Bioportal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 37(Web-Server-Issue):170–173, 2009.
18. A. Tordai, J. van Ossenbruggen, and G. Schreiber. Combining vocabulary alignment techniques. In *K-CAP '09: Proceedings of the fifth international conference on Knowledge capture*, pages 25–32, New York, NY, USA, 2009. ACM.
19. A. Tordai, J. van Ossenbruggen, G. Schreiber, and B. Wielinga. Aligning large SKOS-like vocabularies: Two case studies. In *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 198–212. Springer Berlin / Heidelberg, 2010.
20. P. Vossen, I. Maks, R. Segers, and H. van der Vliet. Integrating lexical units, synsets and ontology in the Cornetto database. In E. L. R. A. (ELRA), editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, 2008.
21. S. Wang, A. Isaac, B. A. C. Schopman, S. Schlobach, and L. van der Meij. Matching multilingual subject vocabularies. In M. Agosti, J. L. Borbinha, S. Kapidakis, C. Papatheodorou, and G. Tsakonas, editors, *ECDL*, volume 5714 of *Lecture Notes in Computer Science*, pages 125–137. Springer, 2009.
22. X. Zhang, Q. Zhong, F. Shi, J. Li, and J. Tang. Rimom results for oaei 2009. In *OM*, 2009.

Chinese Whispers and Connected Alignments

Oliver Kutz[1], Immanuel Normann[2], Till Mossakowski[3], and Dirk Walther[4]

[1] Research Center on Spatial Cognition (SFB/TR 8), University of Bremen, Germany
`okutz@informatik.uni-bremen.de`
[2] Department of Linguistics and Literature, University of Bremen, Germany
`normann@uni-bremen.de`
[3] DFKI GmbH and SFB/TR 8 Spatial Cognition, University of Bremen, Germany
`Till.Mossakowski@dfki.de`
[4] Faculty of Informatics, Technical University of Madrid (UPM), Spain
`dwalther@fi.upm.es`

Abstract. This paper investigates the idea to treat repositories of ontologies as interlinked networks of ontologies, formally captured by the notion of a hyperontology. We apply standard matching algorithms to automatically create the linkage structure of the repository by performing pairwise matching. Subsequently, we define a modular workflow to construct combinations of alignments for any finite number of ontologies. This workflow employs and makes interoperable several tools from the ontology engineering world, comprising matching, reasoning, and structuring tools, and supports in particular modular ontology extraction based on alignment, and a study and empirical analysis of (in)consistency propagation in connected alignments (the Chinese Whispers problem).

Keywords: Hyperontologies; Connected Alignments; Modularity; Consistency

1 Introduction and Problem Description

Ontology matching and alignment based on statistical methods is a relatively developed field, with yearly competitions since 2004 comparing the various strengths and weaknesses of existing algorithms.[5] In this paper, we aim at exploring the degrees to which statistical alignment may lead to inconsistency in the merged ontologies. More precisely, we aim at investigating the effects, both theoretically and practically, of **connected alignment**, i.e. aligning several ontologies that match (non-trivially) pairwise. Our general approach is to treat large repositories of ontologies (in the order of hundreds of ontologies) as our starting point to perform pairwise matching in order to obtain an interlinked network of ontologies. Formally, such networks are captured by the notion of a **hyperontology** [6].

Our work in progress is intended to answer questions such as the following:

> *Assuming pairwise alignments are consistent, how, and when, can we align further ontologies (in various orders) before we drift into inconsistency? In particular, how, and when, can we reduce the question of consistency of aligned ontologies and satisfiability of matched concepts to the consistency of aligned sub-ontologies (i.e. modules generated by the matched sub-signatures)?*

---

[5] See `http://oaei.ontologymatching.org/2009/`

We here set up the theoretical background and necessary engineering environment to give meaningful answers to such questions. In our related paper [9], we have studied techniques of information hiding to support the visualisation of the linkage structure and to allow a user to explore the complex networks resulting from pairwise matching on large sets of ontologies. We here focus on the last question mentioned above, namely how to reduce the consistency problem in an aligned network of ontologies to the consistency of merged modules generated by respective alignments, and the corresponding interoperability problem between matching, modularity, and structuring tools. We also study the effects of matching ontologies in different orders by looking at some specific examples.

Synonymy and Alignment as Colimit Computation. An essential part of the matching and alignment process is to relate and identify signature elements from different ontologies (possibly formulated in different ontology languages). Formally, this is captured by the notion of a **signature morphism**.[6] In the case of $\mathcal{OWL}$ ontologies, these are type-preserving symbol mappings of the form $\sigma : \mathsf{Sig}(O_1) \to \mathsf{Sig}(O_2)$, i.e. mapping the signature of O_1 ($= \mathsf{Sig}(O_1)$) to that of O_2, i.e. concepts to concepts, individuals to individuals, and roles to roles.[7] For a fixed ontology language, a signature morphism straightforwardly induces a sentence translation map.

V-Alignments [13] abstractly capture the alignment process for synonymous signature elements. Given ontologies O_1 and O_2, an **interface** (for O_1, O_2)

$$\big\langle \Sigma, \sigma_1\colon \Sigma \longrightarrow \mathsf{Sig}(O_1), \sigma_2\colon \Sigma \longrightarrow \mathsf{Sig}(O_2)\big\rangle$$

specifies that (using informal but suggestive notation)

- concepts $\sigma_1(c)$ in O_1 and $\sigma_2(c)$ in O_2 are identified for each concept c in Σ, regardless of whether the concepts have the same name or not, and
- concepts in $O_1 \setminus \sigma(\Sigma_1)$ and $O_2 \setminus \sigma(\Sigma_2)$ are kept distinct, again regardless of whether they have the same name or not.

The resulting ontology $\mathcal{O}$ is not given a priori, but rather it is computed from the aligned ontologies via the interface. This computation is a **pushout** in the sense of category theory, which in this case is just a disjoint union with identification of specific parts (namely those given through $\langle \Sigma, \sigma_1, \sigma_2 \rangle$).

V-alignments can deal with basic alignment problems such as *synonymy* (identifying different symbols with the same meaning) and *homonymy* (separating (accidentally) identical symbols with different meaning)—see Fig. 1.

Example 1. In Fig. 1, the interface $\langle \Sigma, \sigma_1, \sigma_2 \rangle$ specifies that the two instances of the concept Woman as well as Person and Human are to be identified. This yields two concepts Woman and Human_Being in the push-out ontology $\mathcal{O}$ obtained along the dashed arrows. It also determines that the two instances of Bank are to be understood as homonyms, and thus generates two new distinct concepts. ⊣

---

[6] See e.g. [5] for the general institutional definition.

[7] We use the DL terminology *concept name* and *role* interchangeably with the $\mathcal{OWL}$ terminology *class* and *property*.

Notion such as *polysemy*, however, are typically understood to relate terms that have a different, but *related* meaning, and can thus not be dealt with by simply identifying symbols or keeping them apart.[8] Similarly, [13] raise the criticism that V-Alignments do not cover the case where a concept Woman in O_1 is aligned with a concept Person in O_2: here, the merged ontology should turn Woman into a subconcept of Person.

$$\begin{tikzcd}
 & \{\mathbf{Woman}, \mathbf{River\_Bank}, \mathbf{Financial\_Bank}, \mathbf{Human\_Being}\} \arrow[d, phantom, sloped, "\supseteq"] & \\
 & \mathcal{O} & \\
O_1 \arrow[ur, dashed] & & O_2 \arrow[ul, dashed] \\
\{\mathsf{Woman}, \mathsf{Bank}, \mathsf{Person}\} \arrow[u, phantom, sloped, "\subseteq"] & & \{\mathsf{Woman}, \mathsf{Bank}, \mathsf{Human}\} \arrow[u, phantom, sloped, "\subseteq"] \\
 & \Sigma \arrow[uul, "\sigma_1"] \arrow[uur, "\sigma_2"'] & \\
 & \{\mathsf{Woman\_Woman}, \mathsf{Person\_Human}\} \arrow[u, phantom, sloped, "="] &
\end{tikzcd}$$

Fig. 1. V-alignment: merge through interface (dashed arrows are automatically computed via colimits)

Whilst this is not directly possible with pushouts, we are here only interested in matching *synonyms across a network of ontologies*, and for this purpose, V-alignments (and their compositions) are sufficient. Studying more complex alignment operations we leave for future work. We next turn to the problem of aligning several ontologies at once.

Consistency and Chinese Whispers. The game of **Chinese Whispers**[9] is played as follows: n persons are arranged in a certain (typically circular) order such that for each person P_i there is a j such that P_i exchanges a message with P_j. The point of the game is to observe the *distortion of the message* as it travels from P_1 along the communication channel. We here are interested in the effects of playing Chinese Whispers with ontologies, where the pairwise matching replaces the transmission of a message, i.e. the messages being exchanged are of the form: *“O_i and O_j agree that concept C of O_i is synonymous with concept D of O_j”.*

We make the following idealisations concerning ‘matching’ a) we assume that in pairwise matching the order does not matter, i.e. matching O_1 with O_2 yields the same colimit ontology (i.e. alignment) as matching O_2 with O_1[10]; b) matching algorithms are ‘not transitive’, i.e., matching $\langle O_1, O_2\rangle$ and $\langle O_2, O_3\rangle$ and computing the colimit yields, in general, a different result than matching and aligning $\langle O_1, O_3\rangle$,[11] c) we assume that we do not match ontologies with themselves.[12]

---

[8] This problem can be addressed by considering $\mathcal{E}$-connections as a general form of alignment (see [6]).

[9] In the United States, “Telephone” is the most common name for the game. The name “Chinese whispers” reflects the former stereotype in Europe of the Chinese language as being incomprehensible. Although it is sometimes considered offensive in the US, it remains the common British English name for the game and is not generally regarded as being offensive.

[10] Whether or not this holds for actual matching systems is an implementational artefact which we ignore; the assumption is certainly reasonable to make as both ‘agreement’ and ‘synonymy’ are symmetric.

[11] In other words, the composition of the two alignments (the composition operation is easily seen to be associative via a pullback operation, see [13]) will typically not agree with a matcher’s results comparing O_1 and O_3 directly.

[12] Although one would expect to get the identity matching in such a case, actual matching tools behave sometimes rather unpredictable in these cases and often return no results.

With these assumptions in place, given a repository $\mathfrak{R}$ with N ontologies, we start with $l = \frac{N \times (N-1)}{2}$ matching pairs.

Playing chinese whispers on $\mathfrak{R}$ with $k \leq N$ players now means to pick a connected subgraph of the hyperontology graph (to ensure that each ontology 'talks' to at least one other), which we call a **matching configuration** . Note that, by assumption, matching configurations contain no loops (i.e. reflexive vertices), are undirected (because of the assumed symmetry of the matching results), and contain at most one edge between two vertices. Therefore, matching configurations are **connected simple graphs**.

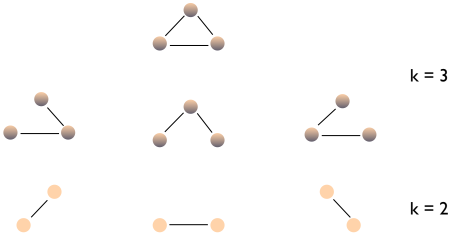


Fig. 2. The number of non-isomorphic matching configurations for $k = 2, 3$

Given a fixed k, the largest possible matching configuration (measured in pairs of matched ontologies, i.e. edges) corresponds to a clique with k nodes, i.e. a complete subgraph of the hyperontology graph with k nodes: such a graph has $\sum_{i=1}^{k-1} i$ edges.

In practise, not every pair of ontologies will match, i.e. a matcher will report no synonyms. Therefore, for fixed $k \leq N$ ontologies $O_1, \ldots, O_k$, the number of non-isomorphic matching configurations containing the O_i $(i = 1, \ldots, k)$ corresponds to the number of connected components of the hyperontology graph with these ontology nodes. The cases of $N = 2, 3$ (assuming a clique) are illustrated in Fig. 2. The alignment operation on a matching configuration, i.e. the computation of the colimit of that graph, we call a **connected alignment**. The **Chinese Whispers Problem** is now the following question:

> *For what kinds (or shapes) of matching configurations and exchanged matching results does the consistency of the input ontologies propagate to the merge (colimit) of the matching configuration?*

We will give some partial answers to this problem in Sec. 2 by showing that the consistency of an aligned matching configuration is reducible to the consistency of the alignment of 'reasonably' large modules talking only about the matched signatures. In Sec. 3, we will discuss in detail an alignment of three ontologies involving the DOLCE ontology, with pairwise consistent alignments, but an overall inconsistent one. Moreover, in Sec. 4, we will describe how the results of a matcher comparing ontologies O_1 and O_2 (and giving rise to a V-alignment) can be rewritten into a structured ontology for further processing with our tool HETS introduced below, and describe the workflow employing standard ontology matching and reasoning tools. Finally, Sec. 5 describes related and future work.

2 Modularity in Hyperontologies

In the following, we will make precise what we mean by a module and define the notion of conservativity. We start with some auxiliary notions. Let Σ be a signature containing concept names and roles. Let $\mathbf{Sen}(\Sigma)$ be the set of sentences formulated using the symbols in Σ in some language. An ontology O in signature Σ is then simply a subset of $\mathbf{Sen}(\Sigma)$. The sentences of course depend on the language the ontologies under consideration are formulated in, e.g. $\mathcal{OWL}$ or some fragment thereof.

We continue with introducing a general notion of a module in the sense that a module of an ontology is not restricted to be a subset of the ontology. It is crucial, however, that the module says everything (expressible in its signature) that is said by the ontology itself (i.e. the ontology is required to be a conservative extension of the module)—see Fig. 3 .

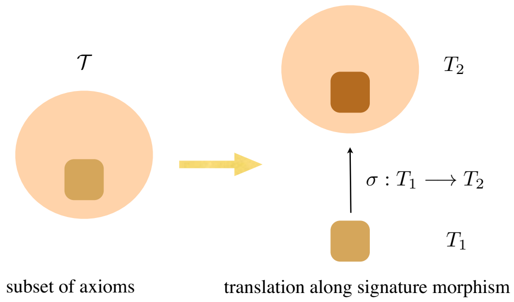


Fig. 3. Modules as subsets vs. modules as image under translation.

Definition 1. *A theory morphism* $\sigma\colon O_1 \longrightarrow O_2$ *is* ***consequence-theoretically conservative****, if* O_2 *does not entail anything new w.r.t.* O_1*, formally,* $O_2 \models \sigma(\varphi)$ *implies* $O_1 \models \varphi$*. Moreover,* $\sigma\colon O_1 \longrightarrow O_2$ *is* ***model-theoretically conservative****, if any* O_1*-model* M_1 *has a* σ*-expansion to* O_2*, i.e. a* O_2*-model* M_2 *with* $M_2|_\sigma = M_1$*.*

Here, $\models$ as usual denotes logical consequence, whereas $\_|_\sigma$ denotes model reduct for a signature morphism $\sigma\colon \Sigma_1 \to \Sigma_2$, i.e. for a Σ_2-model M_2, $M_2|_\sigma$ is a Σ_1-model that interprets a symbol by first translating it along σ and then interpreting it using M_2.

It is easy to show that conservative theory morphisms compose. Moreover, the notion of model-theoretic conservativity is stronger than consequence-theoretic conservativity. To be precise, the former implies the latter, but not vice versa [7]. The two notions coincide if we define consequence-theoretic conservativity using Σ-theories that contain consequences $\phi \in \mathbf{Sen}(\Sigma)$ formulated in second-order logic.

The computational complexity of deciding conservativity appears to be rather daunting even if the ontologies are formulated in weak logics. For instance, for ontologies formulated in the light-weight Description Logic $\mathcal{EL}$, deciding consequence-theoretic conservativity is ExpTime-complete, and model-theoretic conservativity is undecidable. The former problem also becomes undecidable when adding nominals to $\mathcal{ALCIQ}$, for which it is still 2-ExpTime-complete [8]. This suggests that, for practical purposes and applications, we often have to live with approximations of these notions, more precisely with sufficient (syntactic) conditions for conservativity that allow to construct non-minimal modules. Indeed, the notion of an ontology module of an ontology O has been defined as any "subontology O' such that O is a conservative extension of O'" [1].

Definition 2 (Module Generator). *Let O be an ontology in some fixed DL, and let $\Sigma \subseteq \mathsf{Sig}(O)$ be a signature.* $\mathbf{Sen}(\mathsf{Sig}(O))$ *is the set of sentences in* $\mathsf{Sig}(O)$*. A function*

$$\Pi : \langle O, \Sigma \rangle \mapsto \mathbf{Sen}(\mathsf{Sig}(O))$$

mapping pairs $\langle O, \Sigma \rangle$ consisting of an ontology O together with a signature Σ to a set of sentences in $\mathsf{Sig}(O)$ *is called a* ***Σ-module generator*** *if for all O and Σ:*

$$\Pi(\langle O, \Sigma \rangle) \textit{ is a model-theoretic } \mathsf{Sig}(O)\textit{-conservative extension of } O.$$

For a Σ-module generator Π, the set $\Pi(\langle O, \Sigma \rangle)$ is called a ***Σ-module for O****. $\Pi(\langle O, \Sigma \rangle)$ is called* ***Σ-covering for O*** *if:*

$$O \textit{ is a model-theoretic } \Sigma\textit{-conservative extension of } \Pi(\langle O, \Sigma \rangle).$$

The idea to use (conservative) module generators is to massively reduce the size of a colimit ontology to a merge of modules generated by the matched signatures and preserving the semantics completely. This means that we can check the satisfiability of our matched concepts (and the consistency of the overall merged ontology) already in a rather small fragment of the overall ontology. Indeed, it is not hard to construct (or find) cases where ontologies have a moderate semantic overlap, but where the overall merge will be very hard to process by current tools.

Before coming to our central theoretical result, we need some preparatory notions.

Definition 3. *A* ***diagram*** *is a graph D of signatures $(D_i)_{i \in D}$ and signature morphisms $(D_m \colon D_i \to D_j)_{m \colon i \to j \in D}$. Given a diagram D, a family of models $M_i \in \mathbf{Mod}(D_i)_{i \in D}$ is* ***compatible****, if $M_j|_{D_m} = M_i$ for any $m \colon i \to j \in D$. A logic has the* ***amalgamation property*** *w.r.t. a class of diagrams if for any diagram in the class, any compatible family of models can be amalgamated to a unique model of the colimit of the diagram (i.e. such that the reducts along the colimit injections yields the models of the family).*

A logic is ***semi-exact****, if it has the amalgamation property for pushout diagrams.*

Note that colimits of theories can be easily defined in terms of signature colimits and unions of (translated) axioms; the amalgamation property then carries over from signature colimits to theory colimits (see [11], 4.4.17). We formulate the next results in their full generality to cover also ontology languages other than DLs. But note that they apply in particular to all usual DLs as these are semi-exact and moreover have an initial signature (i.e. a signature with a unique signature morphism into any signature) [5].

Theorem 1 (Combination of two modules). *Assume a semi-exact logic. Consider Fig. 4, and assume that $\widetilde{\mathcal{O}}$ and $\mathcal{O}$ are obtained by pushouts (with base Σ). Then $\mathcal{O}$ is a model-theoretic conservative extension of $\widetilde{\mathcal{O}}$. In particular, $\mathcal{O}$ is satisfiable iff $\widetilde{\mathcal{O}}$ is.*

$$\begin{tikzcd}
 & & \mathcal{O} & & \\
O_1 \arrow[urr, dashed] & & & & O_2 \arrow[ull, dashed] \\
 & & \widetilde{\mathcal{O}} \arrow[uu, dashed, "c?"'] & & \\
\Pi(O_1,\Sigma_1) \arrow[uu, "c"'] \arrow[urr, dashed] & & & & \Pi(O_2,\Sigma_2) \arrow[uu, "c"'] \arrow[ull, dashed] \\
 & & \Sigma \arrow[ull, "\sigma_1"] \arrow[urr, "\sigma_2"'] & &
\end{tikzcd}$$

Fig. 4. Propagation of modular structure through one matching (a 'c' denotes conservativity)

Proof. Let $\widetilde{M}$ be a model of $\widetilde{\mathcal{O}}$. For $i = 1, 2$, let M_i be an O_i-expansion of the $\Pi(O_1, \Sigma_i)$-reduct of $\widetilde{M}$ (which exists by conservativity of O_i over $\Pi(O_1, \Sigma_i)$). M_1 and M_2 obviously agree on Σ, hence they have an amalgamation M which is an $\mathcal{O}$-model. Now the $\Pi(O_1, \Sigma_i)$-reducts of M agree with those of $\widetilde{M}$, hence by uniqueness of amalgamation, the $\widetilde{\mathcal{O}}$-reduct of M is $\widetilde{M}$. $\dashv$

We next generalise this result to the case of arbitrary matching configurations.

Theorem 2 (Combination of multiple modules). *Assume a semi-exact logic with an initial signature. Consider a family of ontologies $(O_i)_{i \in I}$ indexed by a finite non-empty set I and a simple graph $G \subseteq I \times I$, such that for $(i, j) \in G$, O_i and O_j are interfaced by*

$$\begin{tikzcd}
O_i & \Sigma_{i,j} \arrow[l, "\theta_{i,j}"'] \arrow[r, "\theta_{j,i}"] & O_j
\end{tikzcd}$$

Define

$$\Sigma_i := \bigcup_{j \in I \setminus \{i\}} \theta_{i,j}(\Sigma_{i,j}) \tag{1}$$

and $\sigma_i \colon \Sigma_i \to \Pi(O_i, \Sigma_i)$ the module in O_i for Σ_i. Let $\sigma_{i,j} \colon \Sigma_{i,j} \to \Pi(O_i, \Sigma_i)$ be the restriction of $\theta_{i,j}$, namely $\theta_{i,j} \colon \Sigma_{i,j} \to \Sigma_i$ composed with σ_i.[13] *Assume that $\widetilde{\mathcal{O}}$ (resp. $\mathcal{O}$) is obtained by the colimit of the diagram of all $\sigma_{i,j}$ (resp. all $\sigma_{i,j}$ composed with the inclusion of $\Pi(O_i, \Sigma_i)$ into O_i). Then $\mathcal{O}$ is a model-theoretic conservative extension of $\widetilde{\mathcal{O}}$. In particular, $\mathcal{O}$ is satisfiable iff $\widetilde{\mathcal{O}}$ is.*

Proof. Note that the diagrams for obtaining $\widetilde{\mathcal{O}}$ resp. $\mathcal{O}$ can be turned into connected diagrams by adding the inclusions of the empty signature into all involved theories (the colimit does not change by this addition). The rationale is that this ensures in $\mathcal{OWL}$ that all compatible model families are built over the same universe of individuals. More formally, by Prop. 4.4.15 of [11], in any semi-exact logic with an initial signature, all finite non-empty connected diagrams enjoy the amalgamation property. With this, the proof is a straightforward generalisation of the proof of Thm. 1. $\dashv$

---

[13] Consider Fig. 4, but replace Σ by $\Sigma_{i,j}$, σ_1 by $\sigma_{i,j}$, σ_2 by $\sigma_{j,i}$, O_1, Σ_1 by O_i, Σ_i, and O_2, Σ_2 by O_j, Σ_j.

Note that Thm. 1 does not hold for consequence-theoretic conservativity. Consider the following example, adapted from [7]. In $\mathcal{ALCO}$, let O_1 be

$$\begin{aligned}\mathsf{IntroTCS} &\sqsubseteq \exists\mathsf{has\_subject}.\mathsf{AutomataTheory}\\ \mathsf{IntroTCS} &\sqsubseteq \exists\mathsf{has\_subject}.\mathsf{ComplexityTheory}\\ \mathsf{AutomataTheory} \sqcap \mathsf{ComplexityTheory} &\sqsubseteq \bot\end{aligned}$$

and O_2 be

$$\begin{aligned}\mathsf{IntroTCS} &\sqsubseteq \forall\mathsf{has\_subject}.\{\mathsf{moore\_automata}\}\\ \mathsf{IntroTCS} &\sqsubseteq \exists\mathsf{has\_subject}.\{\mathsf{moore\_automata}\}\end{aligned}$$

Let Σ be $\{\mathsf{IntroTCS}, \mathsf{has\_subject}\}$. Then assuming a consequence-theoretically conservative (minimal) module generator, $\Pi(O_1, \Sigma) = \Pi(O_2, \Sigma) = \widetilde{\mathcal{O}}$ is

$$\mathsf{IntroTCS} \sqsubseteq \exists\mathsf{has\_subject}.\top$$

But this is consistent, while $O_1 \cup O_2$ is not (assuming $\mathsf{IntroTCS}$ is also instantiated).

3 Worked Out Example

The following worked out example serves two purposes: it should illustrate the last theorem on combination of multiple modules, and it should demonstrate its practical impact.

Our ontology repository ORATE[14] contains among others three ontologies, namely: `SpatialRelations`, `ExtendedDnS`, and `FunctionalParticipation`. Let us denote them by O_1, O_2, and O_3 resp., to be notationally conform with Theorem 2 above. We combine them in different ways and show how consistency issues of the combined ontologies can already be answered in the combination of modules. Automated matching resulted in the concept identifications listed in the two tables below.

`SpatialRelations`	endurant	participant	place-of
`ExtendedDnS`	physical-endurant	participant-place-of	situation-place-of

Table 1. Matching `SpatialRelations` against `ExtendedDnS`

Our first matching thus induces an interface $\langle \Sigma_{12}, \theta_{12}, \theta_{21} \rangle$ with

$$\Sigma_{12} = \{\mathsf{endurant}, \mathsf{participant}, \mathsf{place\text{-}of}\},$$

$\theta_{12} = id$, and $\theta_{21} = \{\text{endurant} \mapsto \text{physical-endurant}, \text{participant} \mapsto \text{paticipant-place-of}, \text{place-of} \mapsto \text{situation-place-of}\}$.

The second an interface $\langle \Sigma_{23}, \theta_{23}, \theta_{32} \rangle$ with $\Sigma_{23} = \{\text{endurant}, \text{physical-object}, \text{region}\}$, $\theta_{23} = \{\text{endurant} \mapsto \text{non-physical-endurant}, \text{physical-object} \mapsto \text{agentive-physical-object}, \text{region} \mapsto \text{space-region}\}$, and $\theta_{32} = id$.

From Σ_{12} and Σ_{23}, the Σ_i's can be determined: $\Sigma_1 = \theta_{12}\Sigma_{12} = \{\text{physical-endurant}\}$, $\Sigma_2 = \theta_{12}\Sigma_{12} \cup \theta_{23}\Sigma_{23} = \{\text{endurant}\}$, and $\Sigma_2 = \theta_{23}\Sigma_{32} = \{\text{non-physical-endurant}\}$.

---

[14] `http://ontologies.informatik.uni-bremen.de`

`SpatialRelations`	endurant	physical-object	region
`FunctionalParticipation`	non-physical-endurant	agentive-physical-object	space-region

Table 2. Matching `SpatialRelations` against `FunctionalParticipation`

The signatures Σ_i, for $i = 1, 2, 3$, together with corresponding ontologies O_i are sent to the module generator that gives us for each ontology the corresponding module $M_i := \Pi(O_i, \Sigma_i)$. Finally, from the signatures Σ_i and the signature morphisms θ_{ij}, the colimit (i.e. the alignment) $\tilde{M}$ of the modules M_1, M_2, and M_3 is obtained. Similarly, the aligned ontology $\tilde{M}$ can be determined from the three ontologies O_1, O_2, and O_3. A practical result of this alignment can be a check of the merged module $\tilde{M}$ for consistency. As we would expect, the merge of the concept "physical-endurant", "endurant", and "non-physical-endurant" into a single concept makes it unsatisfiable—this result can be automatically verified by a prover. Since we know that $\tilde{O}$ must be a conservative extension of $\tilde{M}$, it is in fact not necessary to build $\tilde{O}$ and check its consistency: this information can already be inferred from $\tilde{M}$. To repair the detected inconsistency of $\tilde{M}$, we can abandon either M_1 or M_3 from the alignment process. In both cases, M_1 aligned with M_2 only, or M_3 aligned with M_2 only, the resulting merged module turns out to be consistent, and we know by conservativity that the corresponding ontologies would be consistent, too.

4 An Interoperability Workflow and Prototypical Implementation

4.1 The Component Tools

We implemented a workflow for aligning arbitrary matching configurations taking advantage of several third-party tools. We here briefly introduce these tools and describe in the subsequent subsection their interoperation. Our ontologies to be matched and aligned are taken from our ontology repository ORATE which is being developed and maintained within the EU-project OASIS[15]. The software is based on BioPortal [10]. As matching system we use Falcon [3] which matches $\mathcal{OWL}$ ontologies by means of linguistic and structural analysis. Falcon can be comfortably used in a batch mode and thus makes it suitable for a pipe workflow. For module extraction as well as consistency checks we use Pellet [12] which in particular makes use of the $\mathcal{OWL}$-API [16]. Finally, we use Hets [17] for the computation of colimits (i.e. alignments). Hets is a parsing, static analysis and proof management tool incorporating various provers and different specification languages, thus providing a tool for heterogeneous specifications.

4.2 Workflow Description

Our workflow of multiple ontology alignment consists of two phases: 1) the preprocessing of the whole repository (ORATE) to a complete list of pairwise **matching records**

---

[15] See `http://www.oasis-project.eu`

[16] See `http://owlapi.sourceforge.net`

[17] See `www.informatik.uni-bremen.de/cofi/hets`

and 2) the alignment of a connected subgraph of the **hyperontology graph**. It is up to user what part of the hyperontology graph the user selects as connected subgraph. The rationale behind this user interaction is to dismiss false matchings produced by the matching system. Different matching configuration consequently lead to different alignment outcomes.

```
preprocess_repository = {
  foreach (o1,o2) in repository
   match_ontologies o1 o2
  end
}

match_ontologies o1 o2 = {
  mapping = falcon o1 o2
  cs1 = extract_concepts_from_o1 mapping
  cs2 = extract_concepts_from_o2 mapping
  s12 = build_interface_signature cs1 cs2
  sm1 = build_interface_signature_morphism s12 cs1
  sm2 = build_interface_signature_morphism s12 cs2
  matching_record = (o1,o2,s12,sm1,sm2)
  if (not_empty mapping) store_in_hyperontology_graph matching_record
}

compute_module_graph hyperontology_subgraph = {
  foreach node in hyperontology_subgraph
    in_edges = edges incident with node
    sig = compute_interface_signature node in_edges
    module = compute_module node sig
  end
  replace nodes in hyperontology_subgraph by modules
}

align_modules hyperontology_subgraph = {
  module_graph = compute_module_graph hyperontology_subgraph
  interfaces = {compute_interface edge | edge in module_graph}
  views = {compute_views edge | edge in module_graph}
  spec = write_alignment_spec modules interfaces views
  alignment = compute_alignment spec % hets
}
```

Fig. 5. Pseudo code of the workflow for multiple ontology alignment

Fig. 5 shows the whole workflow in pseudo code. We are going to explain it now line by line. Procedure `preprocess_repository` takes each pair of ontologies `(o1,o2)` and applies the procedure `match_ontologies` to it, i.e., it matches pairwise all ontologies from the repository. The output of `match_ontologies` is a `matching_record`: the matching system Falcon computes the `mapping` between the two ontologies `o1` and `o2`. From the `mapping` the concepts `cs1` (`cs2`) belonging to ontology `o1` (`o2`) are extracted. Based on the concepts `cs1` (`cs2`), the interface signature `s12` is built and the corresponding signature morphisms `sm1` and `sm2` to the ontologies `o1` and `o2`. The two ontologies, the interface signature, and the two signature morphisms form the `matching_record`. This record can be viewed as a link between two ontologies. We call this network whose nodes are ontologies and whose edges are the matching records *hyperontology graph*. Although all ontologies are matched pairwise, the graph is not complete, i.e., some pairs of ontologies (in practice even the majority) are not linked, namely when the matching system cannot find any mappings.

Once the hyperontology graph of the ontology repository is computed we can choose an arbitrary subgraph of it to compute the colimit of modules implicitly given in this subgraph. For that the procedure `compute_module_graph` takes the hyperontology subgraph and basically replaces its ontologies by modules extracted from them. More precisely, for each ontology (=`node`) it collects all interfaces (=`in_edges`) connected to this node and computes (according to Equation 1 in Thm. 2) a signature (`sig`) that is finally used to compute the `module`. Practically, this last step is delegated to Pellet ($\mathcal{OWL}$-API).

Aligning modules implicitly given in a hyperontology subgraph (cf. `align_modules` procedure) comprises the following steps: we first transform the graph with the just mentioned procedure `compute_module_graph` to the `module_graph`. From the module graph we extract all signatures (=`interfaces`) and signature morphisms (`views`) to the modules. From the modules, the interfaces, and the views we compose a specification document (`spec`) that can be understood by Hets. Finally, Hets computes the colimit (`alignment`) of this `spec`.

5 Related Work and an Outlook

Matching and revision in networks of ontologies seems to be a rather unexplored topic. Although only studying pairs of ontologies, the most closely related work in spirit appears to be [4], where a semi-automatic procedure is presented for the integration of ontologies that involves revision of mappings. This approach is implemented as the Protégé 4 plugin ContentMap. Here, a selected ontology matcher is used to compute mappings between the signatures of two ontologies chosen for integration, the mappings are explicitly internalised as axioms in an $\mathcal{OWL}$-2 ontology, and the result of the integration is then taken to be the (disjoint) union of the original ontologies together with the mapping axioms. The integration is assumed to be successful if a user does not identify unintended logical consequences. This decision is guided by justifications (explanations for entailment) that can automatically be computed, e.g., by ContentMap and the confidence values created by the matcher for the mapping axioms. To eliminate the unintended consequences, a repair plan is created describing which axioms from the original ontologies or the mapping should be removed. It should be noted that such a plan does not always exist and that a desired integration may require the iteration of these steps.

The main differences to our approach are a) we support heterogeneity of ontology languages, b) we do not internalise the mappings but make explicit the structure of the alignment graph, c) we generically support arbitrary matching configurations and use category-theoretic techniques to compute the merged ontologies without duplicating matched signature items. Apart from these differences, the necessity for debugging and revising matchings also applies to our approach, and the proposed techniques can be made fruitful also in this setting.

We could here only scratch the surface of the area of problems related to matching in networks of ontologies. We have laid out the necessary engineering infrastructure to combine matching, structuring, and reasoning tools, and obtained some theoretical results concerning the reduction of consistency checks to merged modules.

However, a lot of open questions remain. For instance, internalising mappings can be extended to internalising the confidence values of mappings building on similarity-based $\mathcal{E}$-connections [2]. Moreover, a full statistical analysis of major ontologies and repositories remains to be done to understand the impact of iterated matching on consistency.

Acknowledgements

Work on this paper has been supported by the DFG-funded collaborative research centre SFB/TR 8 'Spatial Cognition', the EU-funded OASIS project, and by the German Federal Ministry of Education and Research (Project 01 IW 07002 FormalSafe). Dirk Walther is supported by a 'Juan de la Cierva' postdoctoral fellowship.

References

1. Ghilardi, S., Lutz, C., and Wolter, F. Did I Damage My Ontology? A Case for Conservative Extensions in Description Logics. In *Proceedings of KR-06* (2006), pp. 187–197.
2. Hois, J., and Kutz, O. Counterparts in Language and Space—Similarity and $\mathcal{S}$-Connection. In *Proc. of FOIS 2008* (2008), C. Eschenbach and M. Grüninger, Eds., IOS Press, pp. 266–279.
3. Hu, W., and Qu, Y. Falcon-AO: A practical ontology matching system. In *Proc. of WWW-07* (2008), pp. 237–239.
4. Jimenez Ruiz, E., Cuenca Grau, B., Horrocks, I., and Berlanga, R. Ontology Integration Using Mappings: Towards Getting the Right Logical Consequences. In *Proc. of the 6th European Semantic Web Conference (ESWC 2009)* (2009), LNCS, Springer.
5. Kutz, O., and Mossakowski, T. Conservativity in Structured Ontologies. In *18th European Conf. on Artificial Intelligence (ECAI-08)* (Patras, Greece, 2008), IOS Press.
6. Kutz, O., Mossakowski, T., and Lücke, D. Carnap, Goguen, and the Hyperontologies: Logical Pluralism and Heterogeneous Structuring in Ontology Design. *Logica Universalis 4*, 2 (2010). Special issue on 'Is Logic Universal?'.
7. Lutz, C., Walther, D., and Wolter, F. Conservative Extensions in Expressive Description Logics. In *Proc. of IJCAI-07* (2007), M. Veloso, Ed., AAAI Press, pp. 453–458.
8. Lutz, C., and Wolter, F. Conservative Extensions in the Lightweight Description Logic $\mathcal{EL}$. In *Proc. of CADE-07* (2007), Springer, pp. 84–99.
9. Normann, I., and Kutz, O. Ontology Reuse and Exploration via Interactive Graph Manipulation. In *Proc. of the ISWC Workshop on Ontology Repositories for the Web (SERES-2010)* (ISWC-2010, November 7, 2010, Shanghai, China), CEUR-WS.
10. Noy, N. F., Shah, N. H., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Montegut, M. J., Rubin, D. L., Youn, C., and Musen, M. A. Bioportal: A web repository for biomedical ontologies and data resources [demonstration], 2007.
11. Sannella, D., and Tarlecki, A. *Foundations of Algebraic Specifications and Formal Program Development*. Springer Verlag. to appear.
12. Sirin, E., Parsia, B., Cuenca Grau, B., Kalyanpur, A., and Katz, Y. Pellet: A practical OWL DL reasoner. *Journal of Web Semantics 5*, 2 (2007), 51–53.
13. Zimmermann, A., Krötzsch, M., Euzenat, J., and Hitzler, P. Formalizing Ontology Alignment and its Operations with Category Theory. In *Proc. of FOIS-06* (2006), pp. 277–288.

Consistency-driven Argumentation for Alignment Agreement

Cássia Trojahn and Jérôme Euzenat

INRIA & LIG, 655 Avenue de l'Europe, Montbonnot Saint Martin, France
{cassia.trojahn,jerome.euzenat}@inrialpes.fr

Abstract. Ontology alignment agreement aims at overcoming the problem that arises when different parties need to conciliate their conflicting views on ontology alignments. Argumentation has been applied as a way for supporting the creation and exchange of arguments, followed by the reasoning on their acceptability. Here we use arguments as positions that support or reject correspondences. Applying only argumentation to select correspondences may lead to alignments which relates ontologies in an inconsistent way. In order to address this problem, we define maximal consistent sub-consolidations which generate consistent and argumentation-grounded alignments. We propose a strategy for computing them involving both argumentation and logical inconsistency detection. It removes correspondences that introduce inconsistencies into the resulting alignment and allows for maintaining the consistency within an argumentation system. We present experiments comparing the different approaches. The (partial) experiments suggest that applying consistency checking and argumentation independently significantly improves results, while using them together does not bring so much. The features of consistency checking and argumentation leading to this result are analysed.

1 Introduction

Due to the diverse ways of exploring the ontology matching problem, matching systems generally differ in the alignments generated between two ontologies. Some approaches may be better suited for some ontologies, or some tasks, than others. Ontology alignment agreement aims at overcoming the problem of allowing different parties to conciliate their conflicting points of view on alignments. There may be different ways to perform alignment agreement, such as voting or weighting. In this paper, we consider argumentation which offers a more reasoned way to decide which correspondences to preserve.

Argumentation theory has been exploited as a way to support the comparison and selection of correspondences within an alignment process. Correspondences are represented as arguments and argumentation frameworks support the reasoning on their acceptability. This approach has been used in different scenarios. [13] propose an approach for supporting the creation and exchange of different arguments, that support or reject correspondences in the context of agent communication. In [18], different matchers work on the basis of particular approaches achieving distinct results that are compared and agreed via an argumentation process.

An open issue in alignment agreement is related to the inconsistency in the agreed alignment. Indeed, some selected sets of correspondences may relate the ontology in an inconsistent way. Most matching systems do not consider logic-based semantics in their algorithms. As a result, almost all matching systems produce incoherent alignments [14]. Although argumentation aims at resolving conflicts on the alignments generated by these systems, this process does not guarantee that the agreed alignment is consistent even if the initial alignments were consistent.

In this paper, we propose a model that involves both argumentation and logical inconsistency detection. We focus on the scenario where matchers working on the basis of different matching approaches try to reach a consensus on their alignments. First, matchers generate their correspondences, representing them as arguments. Next, they exchange their arguments and interpret them under argumentation frameworks based on their individual preferences. The arguments in every individual set of acceptable arguments are considered as an agreed alignment. Then, the inconsistent correspondences in such sets are removed, in order to generate a maximal consistent agreed alignment. This allows for maintaining the consistency within an argumentation system. We evaluate our proposal on a standard set of alignments. Though theoretically grounded, the consistency step does not improve argumentation alone. For some test cases, the argumentation process is incidentally able to provide consistent agreed alignments. We describe the features of consistency checking and argumentation which cause this result.

The rest of the paper is organised as follows. First, we introduce alignments and inconsistency of alignments (§2). We then present the argumentation approach for alignment agreement (§3). Next, the consistency-driven argumentation protocol is presented (§4) and its evaluation is discussed (§5). Finally, we discuss related work (§6) and conclude the paper (§7).

2 Alignments and Inconsistency

An alignment (A) is a set of correspondences from a pair of ontologies (o and o'). Each correspondence is a quadruple: $\langle e, e', r, n\rangle$, where $e \in o$, $e' \in o'$, r is the relation between e and e', taken from set of alignment relations (e.g., $\equiv$, $\sqsubseteq$, $\sqsupseteq$ or $\perp$), and $n \in [0\ 1]$ is a confidence level (e.g., measure of confidence in the fact that the correspondence holds). For instance, given the two ontologies of Figure 1, one can consider the following correspondences, meaning that (1) the two classes Person in both ontologies are the same, and that (2) DepartmentHead in the first ontology is subsumed by Department in the second ontology.

$$\langle \mathsf{Person}_o, \mathsf{Person}_{o'}, \equiv, 1.0\rangle \tag{1}$$

$$\langle \mathsf{DepartmentHead}_o, \mathsf{Department}_{o'}, \sqsubseteq, 0.8\rangle \tag{2}$$

The semantics of alignments provides a definition of how alignments must be interpreted. It is related to the semantics of the aligned ontologies, which is given by their sets of models $\mathcal{M}(o)$ and $\mathcal{M}(o')$. The main effect of alignments is to select compatible pairs of models of the two related ontologies [8].

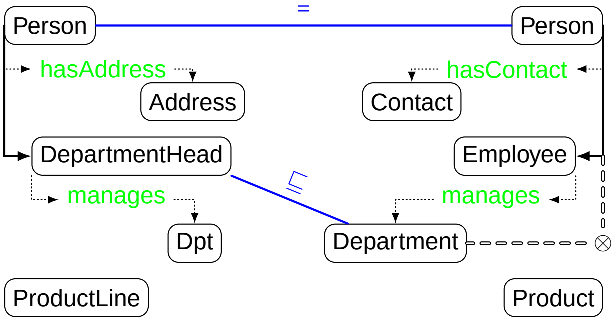


Fig. 1. Fragments of ontologies o and o' with alignment A.

We rely here on a basic semantics in which models are directly compatible. It considers that a correspondence is satisfied by a pair of models if the interpretation of the entities by these models satisfy the relation of the correspondence.

Definition 1 (Satisfied correspondence). *A correspondence $c = \langle e, e', r\rangle$ is satisfied by two models m, m' of o, o' on a common domain $\mathcal{D}$ if and only if $m \in \mathcal{M}(o)$, $m' \in \mathcal{M}(o')$ and*

$$\langle m(e), m'(e')\rangle \in r^U$$

such that $r^U \subseteq \mathcal{D}\times\mathcal{D}$ is the interpretation of the relation. This is denoted as $m, m' \models c$.

For instance, in the language used as example, if m and m' are respective models of o and o':

$$\begin{aligned}
m, m' &\models \langle c, c', \equiv\rangle \text{ iff } m(c) = m'(c')\\
m, m' &\models \langle c, c', \sqsubseteq\rangle \text{ iff } m(c) \subseteq m'(c')\\
m, m' &\models \langle c, c', \sqsupseteq\rangle \text{ iff } m(c) \supseteq m'(c')\\
m, m' &\models \langle c, c', \perp\rangle \text{ iff } m(c) \cap m'(c') = \emptyset
\end{aligned}$$

Definition 2 (Models of aligned ontologies). *Given two ontologies o and o' and an alignment A between these ontologies, a model of these aligned ontologies is a pair $\langle m, m'\rangle \in \mathcal{M}(o) \times \mathcal{M}(o')$, such that each correspondence of A is satisfied by $\langle m, m'\rangle$.*

The alignment acts as a model filter for the ontologies: it selects the interpretation (here the models) of ontologies which are coherent with the alignments. This allows for transferring information from one ontology to another since reducing the set of models will entail more consequences in each aligned ontology.

The notion of models of aligned ontologies is also useful for defining the usual notions of consistency or consequence.

Definition 3 (Consistent alignment). *Given two ontologies o and o' and an alignment A between these ontologies, A is consistent if there exists a model of A. Otherwise A is inconsistent.*

For instance, under the classical ontology interpretation, the alignment A presented in Figure 1 is inconsistent as soon as there exists a DepartmentHead because any model would require to satisfy the following equations:

$$\begin{aligned} m(\mathsf{Person}_o) &= m'(\mathsf{Person}_{o'}) && A \\ m(\mathsf{DepartmentHead}_o) &\subseteq m'(\mathsf{Department}_{o'}) && A \\ m(\mathsf{DepartmentHead}_o) &\subseteq m(\mathsf{Person}_o) && o \\ m'(\mathsf{Department}_{o'}) \cap m'(\mathsf{Person}_{o'}) &= \emptyset && o' \end{aligned}$$

and the DepartmentHead would need to be in both the interpretation of $\mathsf{Department}_{o'}$ and in that of $\mathsf{Person}_{o'}$.

In this paper we will only consider inconsistency, however, the same applies to incoherence: the fact that a class or relation may necessarily be empty, i.e., which would cause inconsistency if instantiated.

3 Argumentation Approach

In alignment agreement, arguments can be seen as positions that support or reject correspondences. Such arguments interact following the notion of attack and are selected according to the notion of acceptability. These notions were introduced by [6]. In Dung's model, the acceptability of an argument is based on a reasonable view: an argument should be accepted only if every attack on it is attacked by an accepted argument. Dung defines an argumentation framework as follows.

Definition 4 (Argumentation framework [6]). *An Argumentation Framework (AF) is a pair $\langle \mathcal{A}, \ltimes \rangle$, such that $\mathcal{A}$ is a set of arguments and $\ltimes$ (attacks) is a binary relation on $\mathcal{A}$. $a \ltimes b$ means that the argument a attacks the argument b. A set of arguments S attacks an argument b iff b is attacked by an argument in S.*

In Dung's model, all arguments have equal strength, and an attack always succeeds (or successfully attacks). [2] has introduced the notion of preference between arguments, where an argument can defend itself against weaker arguments. This model defines a global preference between arguments. In order to relate preferences to different audiences, [3] proposes to associate arguments to the values which supports them. Different audiences can have different preferences over these values. This leads to the notion of *successful attacks*, i.e., those which defeat the attacked argument, with respect to an ordering on the preferences that are associated with the arguments. This allows for accommodating different audiences with different interests and preferences.

Bench-Capon's framework acknowledges the importance of preferences when considering arguments. However, in the specific context of ontology matching, an objection can still be raised about the lack of complete mechanisms for handling persuasiveness [10]. Indeed, many matchers output correspondences with a strength that reflects the confidence they have in the fact that the correspondence between the two entities holds. These confidence levels are usually derived from similarity assessments made during the matching process. They are therefore often based on objective grounds.

For associating an argument to a *strength*, which represents the confidence that an agent has in some correspondence, [18] has proposed the strength-based argumentation framework, extending Bench-Capon's model:

Definition 5 (Strength-based argumentation framework (SVAF) [18]). *A SVAF is a sextuple $\langle \mathcal{A}, \ltimes, \mathcal{V}, v, \succeq, s \rangle$ such that $\langle \mathcal{A}, \ltimes \rangle$ is an AF, $\mathcal{V}$ is a nonempty set of values, $v : \mathcal{A} \rightarrow \mathcal{V}$, $\succeq$ is the preference relation over $\mathcal{V}$ ($v_1 \succeq v_2$ means that, in this framework, v_1 is preferred over v_2), and $s : \mathcal{A} \rightarrow [0, 1]$ represents the strength of the argument.*

Each audience α is associated with its own argumentation framework in which only the preference relation $\succeq_\alpha$ differs. In order to accommodate the notion of *strength*, the notion of *successful attack* is extended:

Definition 6 (Successful attack [18]). *An argument $a \in \mathcal{A}$* successfully attacks *(or* defeats, *noted $a \dagger_\alpha b$) an argument $b \in \mathcal{A}$ for an audience α iff*

$$a \ltimes b \wedge (s(a) > s(b) \vee (s(a) = s(b) \wedge v(a) \succeq_\alpha v(b)))$$

Definition 7 (Acceptable argument [3]). *An argument $a \in \mathcal{A}$ is* acceptable *to an audience α with respect to a set of arguments S, noted $acceptable_\alpha(a, S)$, iff $\forall x \in \mathcal{A}$, $x \dagger_\alpha a \Rightarrow \exists y \in S; y \dagger_\alpha x$.*

In argumentation, a preferred extension represents a consistent position within a framework, which defends itself against all attacks and cannot be extended without raising conflicts:

Definition 8 (Preferred extension). *A set S of arguments is* conflict-free *for an audience α iff $\forall a, b \in S, \neg(a \ltimes b) \vee a \dagger_\alpha b$. A conflict-free set of arguments S is* admissible *for an audience α iff $\forall a \in S, acceptable_\alpha(a, S)$. A set of arguments S in the VAF is a* preferred extension *for an audience α iff it is a maximal admissible set (with respect to set inclusion) for α.*

In order to determine preferred extensions with respect to a value ordering promoted by distinct audiences, *objective* and *subjective* acceptance are defined [3]. An argument is *subjectively acceptable* if and only if it appears in some preferred extension for some specific audience. An argument is *objectively acceptable* if and only if it appears in all preferred extension for every specific audience. We will call *objective consolidation* the intersection of objectively acceptable arguments for all audiences and *subjective consolidation* the union of subjectively acceptable arguments for all audiences.

3.1 Arguments on correspondences

A way of representing correspondences as arguments within an AF is as follows:

Definition 9 (Argument [13, 17]). *An argument $a \in \mathcal{A}$ is a triple $a = \langle c, v, h \rangle$, such that c is a correspondence,$v \in V$ is the value of the argument and h is one of +,- depending on whether the argument is that c does or does not hold.*

The notion of attack is then defined as follow:

Definition 10 (Attack [13, 17]). *An argument* $\langle c, v, h \rangle \in \mathcal{A}$ *attacks an argument* $\langle c', v', h' \rangle \in \mathcal{A}$ *iff* $c = c'$ *and* $h \neq h'$*.*

For instance, if $a = \langle c, v_1, + \rangle$ and $b = \langle c, v_2, - \rangle$, $a \ltimes b$ and vice-versa (b is the counter-argument of a, and a is the counter-argument of b).

The way arguments are generated differs in each scenario. The strategy in [18], *negative arguments as failure*, relies on the assumption that matchers return complete results. Each possible pair of ontology entities which is not returned by the matcher is considered to be at risk, and a negative argument is generated (h=-).

In this paper, different matchers argue with each others in order to obtain an agreement on their alignments. To do this, each matcher is a different audience. The values in $\mathcal{V}$ correspond to the different matching approaches and each matcher m has a preference ordering $\succeq_m$ over V such that its preferred values are those it associates to its arguments. For instance, consider $V = \{l, s, w\}$, i.e., *lexical*, *structural* and *wordnet-based* approaches, respectively, and three matchers m_l, m_s and m_w, using such approaches. The matcher m_l has as preference order $l \succeq_{m_l} s \succeq_{m_l} w$.

To illustrate the agreement process, consider the alignment A of Figure 1 and two matchers i and j. Both i and j generate the correspondence (1) and j the correspondence (2). The following arguments are then created by i and j:

$$\begin{aligned}
a_{i,1} &: \langle \langle \mathsf{Person}_o, \mathsf{Person}_{o'}, \equiv, 1.0 \rangle, w, + \rangle \\
a_{i,2} &: \langle \langle \mathsf{DepartmentHead}_o, \mathsf{Department}_{o'}, \equiv, 0.5 \rangle, w, - \rangle \\
a_{j,1} &: \langle \langle \mathsf{Person}_o, \mathsf{Person}_{o'}, \equiv, 1.0 \rangle, l, + \rangle \\
a_{j,2} &: \langle \langle \mathsf{DepartmentHead}_o, \mathsf{Department}_{o'}, \sqsubseteq, 0.8 \rangle, l, + \rangle
\end{aligned}$$

After generating their arguments, the matchers exchange their arguments with each other. The matcher i sends to j its arguments $a_{i,1}$ and $a_{i,2}$, and vice-versa. i has a preference ordering $w \succeq_i l$, while j has $l \succeq_j w$. Having the complete set of arguments, the matchers generate their preferred extensions p_i and p_j. For both p_i and p_j, the arguments $a_{i,1}$, $a_{j,1}$ and $a_{j,2}$ are acceptable: $a_{i,1}$ and $a_{j,1}$ are not attacked, while $a_{j,2}$ successfully attacks $a_{i,2}$ because both arguments have opposite values of h but $a_{j,2}$ has highest strength than $a_{i,2}$. So, the set of globally acceptable correspondences, A, contains both (1) and (2). It is the alignment associated with the objective consolidation.

Definition 11 (Alignment associated with an extension). *Given an extension* S *in a* $SVAF$*, the alignment associated with this extensions is:* $A(S) = \{c; \exists \langle c, v, + \rangle \in S\}$*.*

However, this set is not consistent. Due to the fact that DepartmentHead is subsumed by Person in o, and Person and Department are disjoint concepts in o', A is inconsistent as soon as there exists one Department.

4 Consistency-driven Argumentation

Resolving the inconsistency problem in alignment agreement has two possible alternatives: (a) express the inconsistency within the argumentation framework, as in [1, 4]; or

(b) deal alternatively with the logical and argumentative parts of the problem. Integrating the logic within the argumentation framework seems a more elegant solution and it can be achieved straightforwardly when correspondences are arguments and incompatible correspondences can mutually attack each others. However, this works only when two correspondences are incompatible. When the set of incompatible correspondences is larger, the encoding is not so straightforward and may lead to the generation of an exponential amount of argument and attacks.

For that purpose, we define the consistency associated with an extension.

Definition 12 (Consistency). *An extension S is said* consistent *iff its associated alignment $A(S)$ is consistent.*

There are different ways to account for consistency in SVAF. The first one retains only consistent preferred extensions. However, the set of preferred consistent extensions may be empty. A fallback would be to consider maximal preferred consistent sub-extensions.

Definition 13 (Maximal preferred consistent sub-extensions). *A consistent extension S is a* maximal preferred consistent sub-extension *iff there exists a preferred extension S' such that $S \subseteq S'$ and $\forall S''; S \subset S'' \subseteq S'$, S'' is not consistent.*

There may be several such sub-extensions. Another approach, considered here, is to work on consolidations, i.e., the set of objective or subjective arguments.

Definition 14 (Maximal consistent sub-consolidations). *A consistent extension S is a* maximal consistent sub-consolidation *of an (objective or subjective) consolidation S' iff $S \subseteq S'$ and $\forall S''; S \subset S'' \subseteq S'$, S'' is not consistent.*

We propose a consistency-driven protocol that computes the maximal consistent objective sub-consolidations. The algorithm removes the correspondences that introduce inconsistencies into the resulting alignment, for maintaining the coherence within the argumentation system. First, as in Section 3.1, the matchers compute their preferred extension from which the objective consolidation, O, is obtained. Based on O, the maximal consistent sub-consolidations is then determined. It can be generalised to consider subjective consolidation or each preferred extension separately. If the objective consolidation is consistent, then the algorithm returns it. If not, the maximal consistent sub-consolidation S is computed.

For computing S we have used the algorithm proposed by [14] which identifies the minimal sets of incoherent correspondences and removes them from the original alignment. The algorithm is based on the theory of diagnosis, where a diagnosis is formed by the correspondences with lowest confidence degrees that introduce incoherence in the alignment. It partially exploits incomplete reasoning techniques to increase runtime performance, preserving the completeness and optimality of the solution.

5 Experiments

5.1 Dataset, matchers and argumentation frameworks

The proposed approach is evaluated on a group of alignments from the conference track of the OAEI[1] 2009 campaign. The data set consists of 15 ontologies in the domain of conference organisation. They have been developed within the OntoFarm project[2]. We use the subset of these test cases where a reference alignment is available (21 alignments, which corresponds to the alignment between 7 ontologies)[3]. We focus on equivalence correspondences, which are taken into account in the reference alignment, and filter out subsumption correspondences.

We have chosen the alignments generated by the four best matchers that have participated in the 2009 OAEI conference track [7]: AMaker, Aflood, AMext and Asmov.

Each matcher has a SVAF and a private preference order, which is based on the f–measure ordering for all matchers – AMaker (0.57), Aflood (0.52), AMext (0.51) and Asmov (0.47). The highest preferred value of each matcher is the value that it associates to its arguments. For instance, AMaker has as preference ordering: $v_{amaker} \succeq_{amaker} v_{aflood} \succeq_{amaker} v_{amext} \succeq_{amaker} v_{asmov}$, while Asmov has the ordering: $v_{asmov} \succeq_{asmov} v_{amaker} \succeq_{asmov} v_{aflood} \succeq_{asmov} v_{amext}$.

For negative arguments ($h = -$), we use two different strength values. First, we consider that the strength can vary according to the matcher quality (conformance with the reference alignment). We assume that this strength is inversely proportional to the probability that a false positive correspondence is retrieved by the matcher. Such probability can be measured by the fallout of the alignment A, given the reference alignment R. Then, we define str for the matcher m:

$$fallout(A, R) = \frac{\mid A \setminus R \mid}{\mid A \mid}, \quad str_m = 1 - fallout(A_m, R)$$

Second, we use str=1.0, assuming that matchers strongly reject correspondences that they do not found (it could be the case when the information about the matcher quality is not available).

5.2 Results and discussion

We measure precision and recall of the maximal consistent sub-consolidation, S, with respect to the reference alignments. First, we present the results from our approach and next we compare them with the results from each matcher. Figure 2 presents the results from the objective consolidations, O, and from the maximal consistent sub-consolidation, S, for SVAFs with $str = 1$ and fallout-based str.

For SVAF with $str = 1$, argumentation (O) is sufficiently selective for generating consistent objective consolidations. We obtain high precision but low recall. This behaviour is due to several reasons. First, we are using objective consolidations and only

---

[1] Ontology Alignment Evaluation Initiative: `http://oaei.ontologymatching.org/`

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

[3] As in [7], the ontology Iasted is filtered out of our experiments because it causes reasoning problems when combined with other ontologies. Thus, we have 15 test cases.

arguments present in every preferred extension are considered (what leads to an increase in precision). Correspondences being accepted by all matchers have high probability to be consistent. Second, we use $str = 1$ for negative arguments ($h = 1$) and thus a true positive (correct) correspondence with strength lower than 1.0 is successfully attacked by a false negative correspondence with strength 1.0 (what decreases the recall).

Using fallout-based str (Figure 2), we have an opposite behaviour. Argumentation is not able to filter out all inconsistent correspondences. We have low precision and high recall. This occurs because negative arguments are not strong enough for successfully attacking all positive arguments (including the incorrect ones). As a result, many correspondences are selected, what increases the probability for selecting inconsistent correspondences. When applying consistency checking, S, in average, precision slightly increases, while recall decreases. This effect is due the way the algorithm for removing the inconsistencies works. An incorrect (but consistent) correspondence might cause the removal of all conflicting correspondences with lower confidence, and thus some correct correspondences are filtered out.

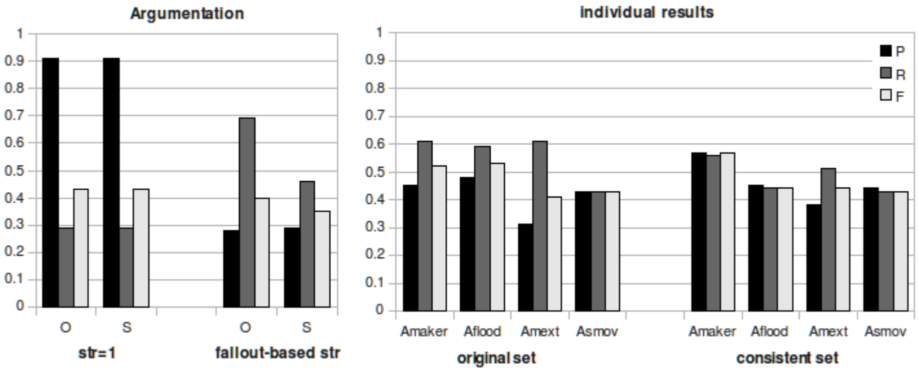


Fig. 2. SVAF with $str = 1$ and fallout-based str: objective consolidation (O) – intersection of objectively acceptable arguments for all audiences, without consistency-checking – and maximal consistent sub-consolidation (S) – consistent subset of objectively acceptable arguments; and individual results for each matcher.

Second, we compare the results from O and S with the results from each matcher. Figure 2 shows the matcher results with and without consistency checking. In the majority of the test cases, the precision increases when filtering out the inconsistent correspondences, while recall decreases (in the case of Aflood, for some tests, the precision decreases while Amaker maintains its recall). As stated before, this is due to the fact that some correspondences are incorrect with respect to the reference alignment but consistent, as well as some correct correspondences are not included in the consistent set because together with some incorrect (but consistent) correspondences, they introduce inconsistencies into the set. Asmov is the only system able to check the consistency in its alignments. In terms of f–measure, apart Asmov, consistency checking improves the results from Amaker and Amext.

Comparing the results from SVAFs with the results from each matcher, for str=1 (Figure 2), argumentation outperforms all matchers in terms of precision, but recall is below all matchers. For fallout-based str, we find an opposite behaviour. All matchers outperform argumentation in terms of precision, but recall is better with argumentation. Looking for argumentation and consistency checking together, although consistency checking slightly improves the precision, both precision and recall are below every matcher. Consistency or argumentation improves results, while contrary to the intuition, we do not observe that the combination of both of these provide more improvements.

Following our (partial) experiments, we can observe that the behaviour of argumentation highly depends on the strength of the arguments. Argumentation is more or less selective when using strong or weak strengths for negative arguments, respectively. Thus, an important issue in the argumentation model is related with the choice of strengths of negative arguments.

Using logical consistency checking alone has positive effects in terms of f–measure for the majority of matchers. On the other hand, combining argumentation and consistency checking slightly improves the precision, when argumentation is not sufficiently selective for generating consistent alignments, but in terms of f–measure, this combination has some negative effects. It is due particularly to the decrease in recall.

6 Related Work

Few ontology matching systems have been developed using semantic-based techniques. Examples of systems using some kind of logical verification are S-Match [9] and ASMOV [11]. S-Match explores propositional satisfiability techniques (SAT) for generating correspondences between graph-like structures. ASMOV semantically verifies the alignments for filtering inconsistencies. However, ASMOV lacks a well defined alignment semantics and notions as correctness or completeness are thus not applicable [14].

In the field of alignment agreement based on argumentation, few approaches have been proposed. In [13], Bench-Capon's model is used to deal with arguments that support or oppose candidate correspondences between ontologies. Both Bench-Capon's and SVAFs frameworks fail at rendering the fact that sources of correspondences often agree on their results, and that this agreement may be meaningful. [10] have adapted the SVAF in order to consider the level of consensus between the sources of the correspondences, by introducing the notions of support and voting into the definition of successful attacks. The work from [5] aims at identifying subparts of ontologies which are sufficient for interpreting messages. This contributes to reduce the consumed time, at a minimal expense in accuracy.

In the field of alignment inconsistency, [15] and [12] considered correcting inconsistent alignments. Revision is obtained exclusively by suppressing correspondences from the alignment through minimising the impact of this suppression. In [15], the goal is to feed the consistent alignment back to a matcher so that it can find new correspondences. This process can be iterated until an eventual fix-point is reached. Similarly, [16] provides a revision operator by modifying one alignment between two ontologies such that the result be consistent. Consistency and consequences are given by merging both ontologies and alignments within the same standard theory. Operators are provided based on the notion of minimal conflict sets.

7 Concluding Remarks

We have defined consistency-driven argumentation for alignment agreement. This fills a gap between argumentation-based matching and consistency-based alignment repairs. We have experimented our strategy on a set of alignments from expressive ontologies. The conclusion is that though theoretically grounded, the extra consistency step does not improve argumentation alone. At least in our experimental setting the argumentation process is incidentally able to provide near consistent extensions. We have analysed the features of consistency checking and argumentation which cause this result.

Hence from these (partial) experiments we can conclude that applying inconsistency recovery and argumentation independently improves results, while using them together does not improve significantly the results. If this does not discard the validity of the approach, it reveals that it should not be applied without care, especially given its complexity.

Further study is required to know better in which context matching and argumentation leads to inconsistency. One source of improvement would be to take into account several such alignments between several ontologies (a network of ontologies). Indeed, these could raise inconsistency within networks of ontologies which would have to be considered as well.

Acknowledgements

We are grateful to Christian Meilicke for letting us use his consistent sub-alignment software.

References

1. L. Amgoud and P. Besnard. Bridging the gap between abstract argumentation systems and logic. In *Proceedings of the 3rd International Conference on Scalable Uncertainty Management*, pages 12–27, Berlin, Heidelberg, 2009. Springer-Verlag.
2. L. Amgoud and C. Cayrol. On the acceptability of arguments in preference-based argumentation. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 1–7, San Francisco, California, 1998. Morgan Kaufmann.
3. T. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.
4. P. Besnard and A. Hunter. Argumentation based on classical logic. In I. Rahwan and G. Simari, editors, *Argumentation in Artificial Intelligence*, pages 133–152. Springer US, 2009.
5. P. Doran, V. Tamma, I. Palmisano, and T. R. Payne. Efficient argumentation over ontology correspondences. In *AAMAS ’09: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 1241–1242, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

6. P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n–person games. *Artificial Intelligence*, 77(2):321–357, 1995.
7. J. Euzenat, A. Ferrara, L. Hollink, A. Isaac, C. Joslyn, V. Malaisé, C. Meilicke, A. Nikolov, J. Pane, M. Sabou, F. Scharffe, P. Shvaiko, V. Spiliopoulos, H. Stuckenschmidt, O. Sváb-Zamazal, V. Svátek, C. T. dos Santos, G. A. Vouros, and S. Wang. Results of the ontology alignment evaluation initiative 2009. In *OM*, volume 551 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.
8. J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, Heidelberg (DE), 2007.
9. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-Match: an algorithm and an implementation of semantic matching. In *Proc. 1st European Semantic Web Symposium (ESWS)*, volume 3053 of *Lecture notes in computer science*, pages 61–75, Hersounisous (GR), 10-12 May 2004.
10. A. Isaac, C. T. dos Santos, S. Wang, and P. Quaresma. Using quantitative aspects of alignment generation for argumentation on mappings. In P. Shvaiko, J. Euzenat, F. Giunchiglia, and H. Stuckenschmidt, editors, *OM*, volume 431 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
11. Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):235 – 251, 2009. The Web of Data.
12. E. Jimenez Ruiz, B. Cuenca Grau, I. Horrocks, and R. Berlanga. Ontology integration using mappings: Towards getting the right logical consequences. In *Proc. 6th European semantic web conference (ESWC), Heraklion (GR)*, volume 5367 of *Lecture notes in computer science*, pages 475–479, 2009.
13. L. Laera, I. Blacoe, V. Tamma, T. Payne, J. Euzenat, and T. Bench-Capon. Argumentation over ontology correspondences in mas. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, New York, NY, USA, 2007. ACM.
14. C. Meilicke and H. Stuckenschmidt. An efficient method for computing alignment diagnoses. In *RR '09: Proceedings of the 3rd International Conference on Web Reasoning and Rule Systems*, pages 182–196, Berlin, Heidelberg, 2009. Springer-Verlag.
15. C. Meilicke, H. Stuckenschmidt, and A. Tamilin. Reasoning support for mapping revision. *Journal of logic and computation*, 19(5):807–829, 2009.
16. G. Qi, Q. Ji, and P. Haase. A conflict-based operator for mapping revision. In *Proc. 8th International Semantic Web Conference (ISWC)*, volume 5823 of *Lecture Notes in Computer Science*, pages 521–536, 2009.
17. C. Trojahn, P. Quaresma, and R. Vieira. Conjunctive queries for ontology based agent communication in mas. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 829–836, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.
18. C. Trojahn, P. Quaresma, R. Vieira, and M. Moraes. A cooperative approach for composite ontology mapping. *LNCS Journal on Data Semantic X (JoDS)*, 4900(0):237–263, 2008.

Alignment-Based Measure of the Distance between Potentially Common Parts of Lightweight Ontologies

Ammar Mechouche, Nathalie Abadie and Sébastien Mustière,

Institut Géographique National, Laboratoire Cogit,
73 Avenue de Paris, 94160 Saint-Mandé, France
{Ammar.Mechouche, Nathalie-f.Abadie, Sebastien.Mustiere}@ign.fr

Abstract. We propose in this paper a method for measuring the distance between ontologies susceptible to describe a common domain and for assessing the feasibility of their integration. This method is in two steps: the first step determines the potentially common parts of two ontologies, based on a prior alignment carried out between them. The second step computes the distance between these parts with regards to both their levels of detail and their structures, by exploiting the mappings contained in the alignment and adapting the Tree Edit Distance method. We limit our study here to lightweight ontologies, i.e., taxonomies represented in OWL[1], the Ontology Web Language. This method was implemented and applied to real ontologies of the geographic domain. The results obtained so far seem significant.

Keywords: Distance between ontologies, Tree Edit Distance, Semantic Web.

1 Introduction

Measuring the distance between heterogeneous ontologies is useful for many applications: 1) retrieving ontologies on the web, e.g. finding an ontology to replace another [8], finding ontologies that can enrich other ones, finding people using same ontologies to create new collaborations, etc.; 2) ontology evolution, in order to know to what extent an ontology, especially its structure, has evolved; 3) ontology fusion and data integration, to know in advance if it may be possible to make joint studies on data described by heterogeneous ontologies.

Ontology matching [7], which is addressed by many works and which consists in computing alignments between ontologies, i.e. determining correspondences between semantically related entities from heterogeneous ontologies, is a key idea enabling interoperability in the semantic web. It has brought solutions to some problems like finding ontologies for query translation. However, for the applications cited above we need to compute global similarity measures between ontologies. Indeed, for example, to enrich an ontology from another one, we need to know if they have close structures and if the second ontology is more detailed than the first one. An alignment between two ontologies does not allow knowing if these latter are complementary or not.

---

[1] http://www.w3.org/TR/owl-features/

In the geographic domain, where data sources are annotated using heterogeneous ontologies [3][5][6][11], we mainly focus on assessing the similarity between ontologies based on three main criteria. The first similarity measure deals with the universes of discourse described by both ontologies: does the source ontology deal with the same domain as the target ontology or does it also provide additional knowledge about a related domain? For example, if we have a domain ontology describing landforms and vegetation, does the source ontology provide us, in addition to that, with knowledge about climate? The second similarity measure aims at comparing the taxonomic structures of both ontologies' common parts. The purpose is to assess whether they result from very different conceptualizations of the domain of interest or not: in other words, would it be difficult to communicate and exchange data with the community who produced this ontology? The third measure compares ontologies' levels of detail to assess whether the source ontology is more or less precise than the target ontology. This aims at automatically determining whether available geodata sources have the appropriate thematic level of detail or not for a specific task. If we are looking for data describing buildings, we may need to make sure that they also explicitly describe more specific buildings such as cabins or huts.

We propose here a new method for computing the distance between potentially common parts of aligned lightweight [17] ontologies. Some works have been dedicated to evaluating the global similarity between ontologies [2], [9], [14-16]. However, their similarity values are difficult to interpret, since they do not measure the difference with regards to particular criteria such as the structure or the level of detail. Our method, however, uses a pre-computed alignment between two ontologies and provides the user with measures with regards to both their structures and levels of detail. This allows to more efficiently assessing differences between ontologies. Moreover, it computes distance between potentially common parts of ontologies, since two ontologies may be similar on one common thematic but quite different on another one. In fact, when source ontologies have common parts with the target ontology, their taxonomic structures must also be compared in order to evaluate to what extent they result from similar conceptualization of the domain of interest. Moreover, the source ontologies' level of detail must be evaluated to assess whether they are more or less precise than the target ontology.

The remainder of this paper is structured as follows: section 2 presents our method for measuring the distance between sub-parts of ontologies. Section 3 presents the results obtained with our method, and section 4 gives some perspectives to our work.

2 Proposed Method for Assessing Differences between Ontologies

Our method, first, uses a simple decision tree and the results of an alignment carried out between the two ontologies to be compared, in order to determine the "important concepts" of each ontology, which encompass a large number of mapped concepts. Once the important concepts are determined, the sub-parts of ontologies whose they are roots are compared. The comparison of ontologies (or ontology parts) here consists in computing the distance with respect to their structures and their levels of detail. To do that, we propose a Tree Edit Distance based method to compute the distance between the structures of the compared ontology parts, and we exploit the results of the alignment performed between them to compute their levels of detail.

2.1 Determining the Important Concepts of two Aligned Ontologies

Our goal here is to determine the potentially common parts of two ontologies. To do that, we use existing tools to align the ontologies we want to compare and then we determine the parts of each ontology where the mappings are concentrated. Indeed, we believe that if two ontologies describe a common sub-domain then the concepts describing this sub-domain in both ontologies should generally be mapped together. We do not try here to improve the quality of the alignment, which is out of the scope of this paper. We suppose that the quality of the produced alignment is pretty good and the mappings it contains could be transformed into the following form: $< C_{O_1}, D_{O_2}, Score >$, which means that the concept *C* in the ontology *O1* is mapped with the concept *D* in the ontology *O2*, and this mapping has a score confidence *Score* (between 0 and 1).

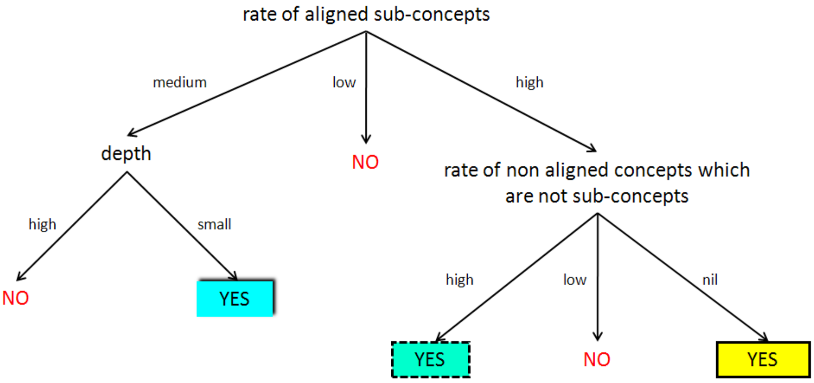


Fig. 1. The decision tree determining the important concepts of two aligned ontologies.

Before detailing our method, we consider that a concept is important if it encompasses a large number of mapped sub-concepts or if it encompasses a medium number of mapped sub-concepts but at the same time its depth in the ontology is small. In both cases, the important concepts define sub-parts of the ontology where the mappings are concentrated.

A first set of important concepts is determined thanks to the decision tree depicted in Fig. 1, which classifies each concept of the ontology in the class *YES* if it is an important concept or in the class *NO* if it is not . After that, additional rules allow to filter the first set of important concepts and to keep only the most significant ones**.**

The decision tree proposes three rules allowing determining important concepts:

The first rule (leading to the *YES* in the continuous and bold rectangle) detects the root of each ontology (the *owl:Thing* concept) as the important concept if the number of all mappings is high, since the root is the super class of any concept of the ontology, and it encompasses all mapped concepts. On the examples shown on Fig. 2 the important concepts deduced by this rule are in continuous and bold circles.

The second rule (leading to the *YES* in the dashed rectangle) allows determining the important concepts which encompass a large number of mapped concepts. This rule does not allow to a concept *C* to be important if there are too few remaining concepts in the ontology that are not mapped and not sub-concepts of *C*. In this case it is better to consider the root of the ontology as the important concept. For example, on Fig. 2-(a1) the concept *3* could be an important concept if we suppose that all mapped concepts are its sub-concepts. However, since there remains only one concept (concept *2*) in the ontology which is not its sub-concept it is better to consider only the root as an important concept in order to include this remaining concept. In Fig. 2(b1), however, there are two important concepts deduced by this rule (those with dashed lines). This is due to the fact that: 1) a large number of the mapped concepts are sub-classes of the concept *3* and consequently sub-classes of its parents ; 2) there remains a large number of concepts in the ontology which are not sub-concepts of *3* and which can constitute another part describing another thematic which is not common to the compared ontologies.

The third rule (leading to the *YES* in the shadow rectangle in the decision tree) allows determining the important concepts which encompass a medium number of mapped concepts and which are not deep in the ontology hierarchy. Indeed, in general, in an ontology, the distinction of the different described themes is made generally in its top level. In the example on Fig. 2(c1) the important concepts deduced by this rule are represented with shadow circles (*2*, *3* etc.). The important concepts are determined by this rule when the compared ontologies describe more than one common thematic.

Until now we determine a set of possible important concepts. For example, on Fig. 2(b1) and Fig. 2(c1) we have several important concepts which are deduced and we need to keep only the most significant among them, i.e. the more specific of them. To do this, we defined three additional rules allowing filtering these important concepts. They are tested in the following order, and only one of them is executed:

Filtering Rule 1: If we have only one important concept (the root), then keep it as the important concept of the ontology. For example, on Fig. 2(a2) the concept *1* will be the important concept of the ontology; in other terms, the source ontology will be compared as a whole to the determined sub-parts of the target ontology.

Filtering Rule 2: If we have important concepts deduced by the second rule of the decision tree (concepts in dashed circles), then we keep only the deepest among them as the important concept of the ontology. Indeed, in an ontology, the deeper the concepts are, the more they share common characteristics. In Fig. 2(b2), the important concept which will be kept is *3* since the Rule 1 will not be activated and the concept *3* is the deepest important concept deduced by the second rule of the decision tree. This way we obtain one important concept encompassing a large number of mapped concepts without including concepts describing other themes.

Filtering Rule 3: We keep all the deepest important concepts deduced by the third rule of the decision tree (concepts in shadow circles). For example, in figure Fig.

2(c2), we keep only the important concepts *2* and *3* since they are the deepest ones. These concepts should be roots of two different themes described by the ontology.

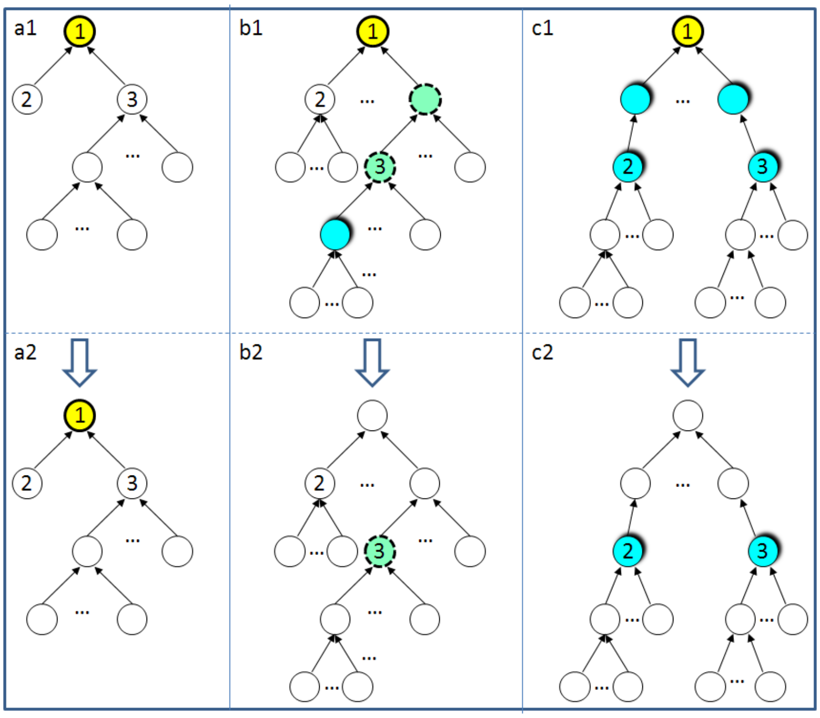


Fig. 2. Possible cases for determining the important concepts.

2.2 Computing the Distance between the Ontology Parts

We consider here that two aligned ontologies *O1* and *O2* have the same structure if the subsumption relations (or *is-a* relations) between all mapped concepts are preserved in both ontologies. In other terms, the ontologies structures are the same if we do not consider the non mapped concepts. If there is one *is-a* relation between two mapped concepts *C1* and *C2* in *O1*, and this relation does not exists in *O2* between the corresponding concept of *C1* and the corresponding concept of *C2*, then we consider that the structures of *O1* and *O2* are different.

On Fig. 3(1) we have two ontologies with the same structure, but with different levels of detail. In this case, it is interesting to indicate to the user that these ontologies have similar structures, but that one is more detailed than the other one. This allows the user to decide, for example, to enrich the first ontology from the second one or to know if their fusion would be costly or not. In figure Fig. 3(2), however, both ontologies offer the same vocabulary, but their structures are different. Typically, in this case it would be costly to fusion or to combine these ontologies.

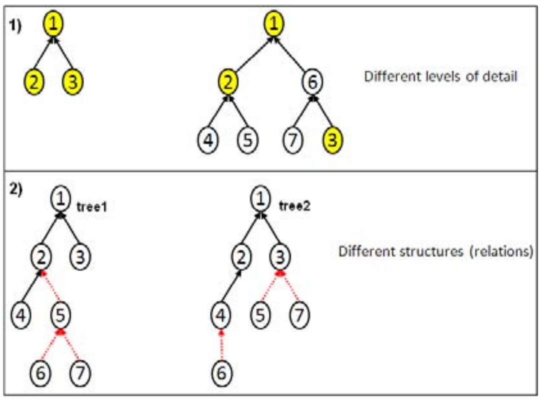


Fig. 3. Difference of levels of detail and structures between ontologies.

2.2.1 Comparing Ontologies' Structures

To compute the distance between the structures of two ontologies, we propose an adaptation of the Tree Edit Distance method [1], which is usually used to estimate the minimum effort which is necessary to transform an ordered tree into another one. We note that an ordered tree is a tree where the children of every node are ordered. The Tree Edit Distance method returns the minimum cost in terms of the number of operations (node insertion, node deletion and node renaming) which are necessary to transform one ordered tree into another one. Let us consider the example on Fig. 3(2). In order to transform *tree1* into *tree2* we need at least five operations (Fig. 4). So, transforming *tree1* into *tree2* costs 5.

In order to give a sense to the cost returned by the Tree Edit Distance method, we need to normalize it. To do this, we use the normalization formula proposed in [4] (formula (2)), where *NC* is the normalized cost considered as the distance between the two trees, *C* is the value returned by the Tree Edit Distance method, and |*tree 1*| and |*tree 2*| are the respective sizes (number of nodes) of the ordered trees *tree1* and *tree2*.

$$NC = \frac{C}{|tree_1| + |tree_2|} \qquad (2)$$

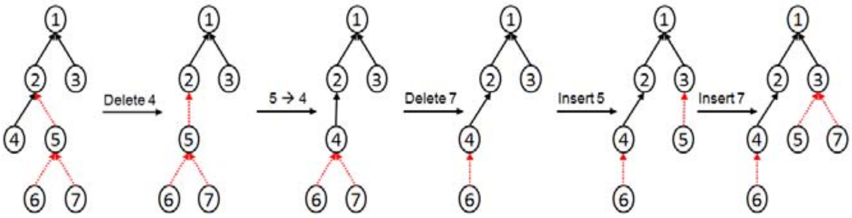


Fig. 4. Example of minimum operation number required to transform a tree into another one.

In the example on Fig. 4 the distance between *tree 1* and *tree 2* will be 5/(7+7) = 0.36, which means that similarity between *tree 1* and *tree 2* is 64%. In fact, the smaller the distance is, the closer the structures of ontologies are.

The adaptation of the Tree Edit Distance method is done as follows:

1. Every non mapped concept of each ontology is deleted and its mapped direct sub-concepts become direct sub-concepts of its closest mapped parent. Indeed, we are only interested in the structures formed by mapped concepts in each ontology. In Fig. 5, the concepts *8* and *9* in the ontology on the left are not mapped with concepts from the ontology on the right, so they are deleted and the concept *3* which is mapped becomes a direct child of the concept *1*. The same reasoning is applied to the concept *g*.

2. The next step consists in relabeling the concepts in the second ontology by the labels of their corresponding concepts in the first ontology. If a concept *C* from the second ontology is mapped with one concept *D* from the first ontology, then *C* is renamed to *D*. For example, on Fig. 5, the concept *a* is mapped with the concept *1*, then *a* is renamed to *1*. If a concept *C* of the second ontology is mapped with several concepts from the first ontology, then it takes the name of the corresponding concept having the highest score of similarity. On the Fig. 5, the concept *h* is mapped with the concepts *6* and *7* from the first ontology*,* however the score of the mapping with *7* is higher, then *h* will be renamed to *7*. If it had been mapped with several concepts from the first ontology having the same score of similarity, then the label of the most general subsuming concept of them would have been used for renaming. If no most general subsuming concept exists, then we would have chosen randomly the label of one of them for renaming. On the Fig. 5, the concept *f* is mapped with the concepts *5* and *6* from the first ontology with the same score, then *f* has to be renamed to *5*, since *5* is more general than *6*. We note that each concept from the first ontology is used at most one time to rename a concept in the second ontology.

3. The last step consists in ordering the obtained trees to allow using existing algorithms and tools for computing the Tree Edit Distance (which is our objective here). On the Fig. 5 we obtain two ordered trees whose distance was computed in the previous example (distance = 0.36).

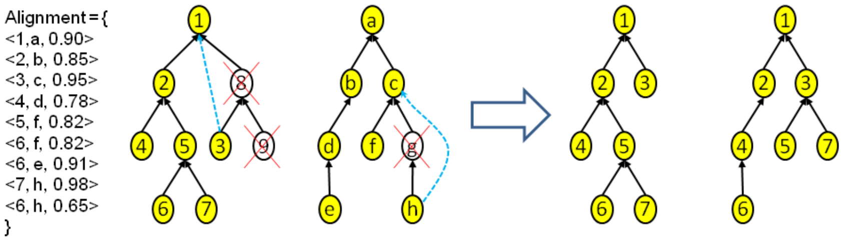


Fig. 5. Transforming two aligned ontologies into two ordered trees.

2.2.2 Computing the Levels of Detail of two Aligned Ontologies

Two lightweight ontologies having close structures may have different levels of detail. We define here the level of detail of a concept *C* as the number of its sub-classes. For example, on Fig. 3(1) the concept *2* is more detailed in the ontology on the right than in the ontology on the left, since it has two sub-concepts in the ontology on the right when it has no sub-concept in that on the left. Now, we need to have an overall indication allowing to know which of the two aligned ontologies is more detailed. We propose to do this by averaging, for each ontology, the levels of detail' values of its mapped concepts. The following example illustrates this more clearly.

Illustrating example. Let us consider the example on Fig. 3(1). To compute the level of detail of each ontology we create two vectors *V1*and *V2*, where *V1* contains the level of detail values of each mapped concept from *O1*, and *V2* contains the level of detail values of the corresponding concept in *O2* of each mapped concept from *O1*. Then, the average value of *V1* gives the level of detail of *O1*, and the average value of *V2* gives the level of detail of *O2*. Thus, we obtain the following vectors:

$$\begin{array}{ccc} Mapped\ \ Concepts & V1 & V2 \\ \begin{bmatrix}1\\2\\3\end{bmatrix} & \begin{bmatrix}2\\0\\0\end{bmatrix} & \begin{bmatrix}6\\2\\0\end{bmatrix} \\ & LD_{O_1}=0.66 & LD_{O_2}=2.66 \end{array}$$

As we can see it, *O2* is more detailed than *O1*.

3 Experiments and Results

The proposed methods were implemented and tested on real ontologies. The decision tree, determining the important concepts, was implemented in Java and using the Protégé OWL API[2]. The ontology structures comparison was also implemented as a Java program which uses the Protégé OWL API and reuses a Java implementation of the Tree Edit Distance method for ordered trees, available on the Web[3] and described in [1]. The goal here is to determine which parts of two aligned ontologies are complementary, i.e. which parts are related by a large number of mappings and which have close structures but different levels of detail. We chose to test our methods on five ontologies (ontologies' structures) describing geographic domains or domains close to geography, because our expertise in this domain allows us to better analyze the obtained results. The used ontologies are the followings:

- Building and Places ontology[4]: developed in United Kingdom, its purpose is to describe the building feature and place classes surveyed by Ordnance Survey.
- Transportation ontology[5]: this ontology describes transportation-related information in the CIA World Fact Book[6].

---

[2] http://protege.stanford.edu/plugins/owl/api/

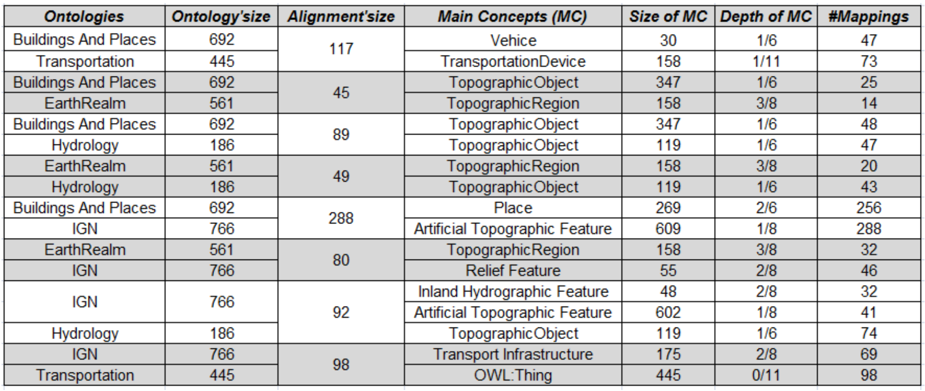
[3] web.science.mq.edu.au/~swan/howtos/treedistance/

[4] http://www.ordnancesurvey.co.uk/ontology/BuildingsAndPlaces/v1.1/BuildingsAndPlaces.owl

[5] http://reliant.teknowledge.com/DAML/Transportation.owl

- Earth Realm ontology[7]: elements of this ontology include “atmosphere”, “ocean”, and “solid earth”, and associated subrealms (such as “ocean floor”)[8].
- Hydrology[9]: this ontology is developed by Ordnance Survey to describe in an unambiguous manner the inland hydrology feature classes.
- IGN ontology [12]: it is a bilingual ontology (French / English) which describes the topographic entities present in the geographic databases of the French Mapping Agency (IGN).

These ontologies are first pairwise aligned using the method proposed in the TaxoMap tool [10]. In order to determine the important concepts we defined the semantics of classification criteria used in the decision tree. So, we consider that the rate (percentage) of aligned concepts is high when it is superior to 80%. It is medium when it is comprised between 30% and 80%, and it is low when it is inferior to 30%. A concept of an ontology *O* is considered as deep if its depth in *O* is higher than a half of the depth of *O* itself. We note that, actually, these values are fixed intuitively, however we are working on in order determine them empirically. The obtained results with these values are summarized on Fig. 6.



Ontologies	*Ontology'size*	*Alignment'size*	*Main Concepts (MC)*	*Size of MC*	*Depth of MC*	*#Mappings*
Buildings And Places	692	117	Vehice	30	1/6	47
Transportation	445		TransportationDevice	158	1/11	73
Buildings And Places	692	45	TopographicObject	347	1/6	25
EarthRealm	561		TopographicRegion	158	3/8	14
Buildings And Places	692	89	TopographicObject	347	1/6	48
Hydrology	186		TopographicObject	119	1/6	47
EarthRealm	561	49	TopographicRegion	158	3/8	20
Hydrology	186		TopographicObject	119	1/6	43
Buildings And Places	692	288	Place	269	2/6	256
IGN	766		Artificial Topographic Feature	609	1/8	288
EarthRealm	561	80	TopographicRegion	158	3/8	32
IGN	766		Relief Feature	55	2/8	46
IGN	766	92	Inland Hydrographic Feature	48	2/8	32
			Artificial Topographic Feature	602	1/8	41
Hydrology	186		TopographicObject	119	1/6	74
IGN	766	98	Transport Infrastructure	175	2/8	69
Transportation	445		OWL:Thing	445	0/11	98

Fig. 6. Important concepts obtained on real geographic ontologies.

We observe on Fig. 6 that most of the used ontologies describe the topography; many important concepts are in relation to the topography, which is a good result since most of the used ontologies really describe the topographic objects. Also, if we look at the size of each determined partition (whose root is an important concept) and the number of its mappings, we deduce that the important concepts detection is pretty precise. For example, the number of mappings between the *Buildings and Places* ontology and the *IGN* ontology equals 288. From one side, there is only one important concept deduced for each ontology (respectively *Place* and *Artificial Topographic Feature*). From the other side, the number of mappings included in the part whose

---

[6] http://www.daml.org/ontologies/409

[7] http://sweet.jpl.nasa.gov/1.1/earthrealm.owl

[8] http://sweet.jpl.nasa.gov/guide.doc

[9] http://www.ordnancesurvey.co.uk/ontology/Hydrology/v2.0/Hydrology.owl

Place is the root represents 89% of the total number of existing mappings, and the part whose *Artificial Topographic Feature* is the root contains all the mappings. So, the mappings are concentrated in the parts determined by our method.

The next step is the comparison phase. In order to obtain significant results we consider only mappings with a score higher than 0.90. We first computed the distance between the ontology parts structures. The obtained results are shown on Fig. 8.

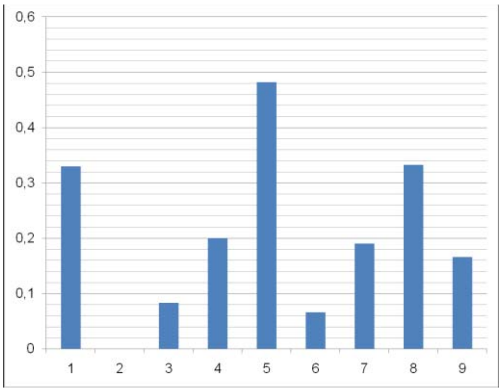
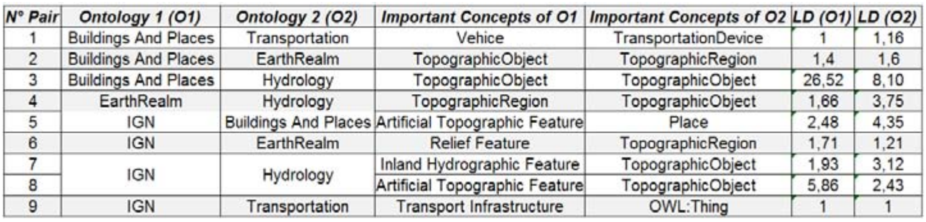


Fig. 8. Results of the comparison of the ontology parts structures. The vertical axis indicates the computed Tree Edit Distance measures, and the horizontal axis indicates the compared ontology parts. Indeed, the numbers 1...9 refer to pairs of ontology parts that are compared. See Fig. 9.

Fig. 8 shows that there are some ontology parts that have more similar structures than other ones. For example, the structure of the part of the *Buildings and Places* ontology whose the root is the important concept *Topographic Object* is very close to the structure of the part of the *Hydrology* ontology that has *Topographic Object* as a root. This is due to the fact that both ontologies are produced by the same institution, so with same conceptualization. The structure of the part of the *Buildings and Places* ontology whose the root is the important concept *Place* is, however, different from the structure of the part of the *IGN* ontology that has *Artificial Topographic Feature* as a root. In fact, in the metadata associated with *Building and Places* ontology, it is said that "…*The rationale behind the Buildings and Places module is to provide a minimal set of definitions to maximise the abiliuty to reuse. As a result it contains a* ***shallow hierarchy*** *and minimal property restrictions*". This explains the difference in structures between its parts and parts of the *IGN* ontology which is very structured.

Finally, we compared the levels of detail (LD) of our ontologies using the method presented above. The results obtained are shown on Fig. 9.



N° Pair	Ontology 1 (O1)	Ontology 2 (O2)	Important Concepts of O1	Important Concepts of O2	LD (O1)	LD (O2)
1	Buildings And Places	Transportation	Vehice	TransportationDevice	1	1,16
2	Buildings And Places	EarthRealm	TopographicObject	TopographicRegion	1,4	1,6
3	Buildings And Places	Hydrology	TopographicObject	TopographicObject	26,52	8,10
4	EarthRealm	Hydrology	TopographicRegion	TopographicObject	1,66	3,75
5	IGN	Buildings And Places	Artificial Topographic Feature	Place	2,48	4,35
6	IGN	EarthRealm	Relief Feature	TopographicRegion	1,71	1,21
7	IGN	Hydrology	Inland Hydrographic Feature	TopographicObject	1,93	3,12
8			Artificial Topographic Feature	TopographicObject	5,86	2,43
9	IGN	Transportation	Transport Infrastructure	OWL:Thing	1	1

Fig. 9. Result of the comparison of the ontology parts levels of detail (LD).

The results on the Fig. 9 are significant. For example, the *Hydrology* ontology is more detailed than the *IGN* ontology regarding the hydrographic features, and this can be explained as follows: from one side the metadata associated with the *Hydrology* ontology say that *the scope of this ontology includes permanent topographic features involved in the containment and transport of surface inland water of a size of **1** meter or greater including tidal water within rivers*. From the other side, we know from the IGN databases specifications that the IGN databases, from which is built the IGN ontology, include only water surfaces larger than **7.5** meter. Another notable difference of levels of detail is between the *Topographic Object* part of the *Hydrology* ontology and the *Topographic Object* part of the *Buildings and Places* ontology. This result combined with the previous one telling us that the structures of these parts are very similar shows that these ontology parts are complementary and may help the user or a program to decide for example to enrich (for example thanks to an importation operation) the *Topographic Object* part of the *Hydrology* ontology from the *Topographic Object* of the *Buildings and Places* ontology.

4 Conclusion and Perspectives

Means for evaluating distance between ontologies seems to us important for decision making systems in the context of data integration, ontology fusion, ontology evolution and ontology retrieval on the web.

We have presented in this paper a new method for measuring the distance between lightweight ontologies. Our method differs from existing methods in several ways: 1) it exploits alignments between ontologies rather than assuming that ontologies share exactly the same vocabulary; 2) it does not compare whole ontologies but only the potentially common parts of them determined by our decision tree, in order to more efficiently assess the differences between ontologies; 3) finally, our method provides indications about the level of detail of each ontology and computes a distance between the ontologies structures by adapting the Tree Edit Distance method, which was not used in the past in this context to the best of our knowledge. The proposed method is implemented and tested on several real geographic ontologies, and the results obtained so far seem significant.

In the future, we plan to improve our method with respect to several aspects: 1) the ontology parts are determined with our decision tree which consists in detecting ontology parts where the mappings are concentrated, it would be interesting to compare and combine our decision tree with existing methods for ontology partitioning [13], in order to obtain better partitions. Moreover, our decision tree may be learnt using existing algorithms like the ID3[10] algorithm, particularly in order to automatically determine the semantics associated with each classification criteria; 2) our comparison method is actually restricted to lightweight ontologies, we plan to extend it to heavyweight ontologies in order include in our comparison procedures more complex constraints and relations between concepts modeled in domain ontologies; 3) another perspective of this work is to compare our method to similar ones [18] and to study the influence of different matching techniques on our distance measure; 4) finally, we plan to integrate to our method other information to better

---

[10] http://en.wikipedia.org/wiki/ID3_algorithm

understand differences between ontologies, like metadata associated with ontologies, ontology utilization purposes, etc.

Acknowledgement

This work is partly founded by the French Research Agency through the GeOnto project ANR-O7-MDCO-005 on “creation, alignment, comparison and use of geographic ontologies”.

References

1. Zhang K., Shasha D.: Fast algorithms for the unit cost editing distance between trees. Journal of Algorithms, 11(4), pp. 581-621, (1990)
2. Maedche A., Staab S.: Measuring Similarity between Ontologies. EKAW, 251-63, (2002)
3. Rodriguez M.A, Egenhofer M.J.: Determining Semantic Similarity Among Entity Classes from Different Ontologies. IEEE TKDE, 15(2), p. 442-56. (2003)
4. Sager T., Bernstein A., Pinzger M., Kiefer C.: Detecting similar Java classes using tree algorithms. International Workshop on Mining Software Repositories, 65-71, (2006)
5. Paul, M., Ghosh S.K.: An Approach for Service Oriented Discovery and Retrieval of Spatial Data, Procedings International Workshop on Service Oriented Software Engineering (IW-SOSE’06), Shangay, China, pp 84—94, (2006)
6. Nambiar U., Ludäscher B., Lin K. and Baru, C.: The GEON portal: accelerating knowledge discovery in the geosciences, 8th ACM International Workshop on Web Information and Data Management, Aarlington, pp 83—90, (2006)
7. Euzenat J. and Shvaiko P.: Ontology Matching. Springer Verlag, p. 333, (2007)
8. Jérôme David, Jérôme Euzenat: Comparison between Ontology Distances (Preliminary Results). International Semantic Web Conference, 245-260,(2008)
9. Wang J.Z., Ali F. and Srimani P.K: An Efficient Method to Measure the Semantic Similarity of Ontologies. Advances in Grid and Pervasive Computing: p. 447-458, (2008)
10. Hamdi F., Zargavouna H., Safar, B. and Reynaud C.: TaxoMap in the OAEI 2008 alignment contest. Ontology Matching, p. 206-213, (2008)
11. Klien E.M.: Semantic Annotation of Geographic Information. Phd thesis, Institute for Geoinformatics, University of Muenster. Muenster, Germany, (2008)
12. Abadie N. and Mustière S.: Constitution d’une taxonomie géographique à partir des spécifications de bases de données. Actes de SAGEO, Montpellier, (2008)
13. Stuckenschmidt H., Parent C. and Stefano Spaccapietra (Eds.): Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization. LNCS Vol. 5445 Springer, ISBN 978-3-642-01906-7, (2009)
14. Euzenat J., Allocca C., David J., d'Aquin M., Le Duc C. and Sváb-Zamazal, O.: Ontology distances for contextualisation. Report 3.3.4, IST NeOn IP, NeOn, (2009)
15. Ngan L., Soong A. and Hung L.: Comparing two ontologies. International Journal on Web Engineering and Technologies, 5(1): p. 48-68, (2009)
16. Mechouche A., Abadie N., Mustière S. : Mesure de la distance sémantique entre parties potentiellement communes à deux taxonomies, *IC*, France, (2010)
17. Óscar Corcho: Ontology based document annotation: trends and open research problems. IJMSO 1(1): 47-57 (2006)
18. P. Wang, B. Xu. Lily: Ontology Alignment Results for OAEI 2009. OM (2009)

Mapping the Central LOD Ontologies to PROTON Upper-Level Ontology

Mariana Damova, Atanas Kiryakov, Kiril Simov, Svetoslav Petrov

[1] Ontotext AD, Tsarigradsko Chosse 135, Sofia 1784, Bulgaria
{ mariana.damova,naso,kivs,svetoslav.petrov}@ontotext.com

Abstract. Linking Open Data (LOD) facilitates the emergence of a web of linked data by publishing and interlinking open data on the web in RDF. One can explore linked data across servers by following the links in the graph. The LOD cloud has 203 datasets and more than 14 billion RDF triples (http://lod-cloud.net). This paper describes an approach to access these data by means of a single ontology, matched to the schemata describing several of the most common LOD datasets. They are presented in a reason-able view - FactForge (http://factforge.net) - the biggest and most heterogeneous body of factual knowledge on which inference is performed. Techniques of (a) making matching rules with "ontology expressions", (b) adding new instances with inference rules, and (c) extending the upper level ontology with classes and properties are employed. They succeed to align ontologies designed according to different principles and displaying conceptual and structural mismatches.

Keywords: Linked Open Data, FactForge, PROTON, ontology matching, upper level ontology, semantic web, RDF, dataset, DBPedia, Freebase, Geonames.

1 Introduction

Linking Open Data (LOD) initiative [1] aims to facilitate the emergence of a web of *linked data* by means of publishing and interlinking open data on the web in RDF. One can explore linked data across servers by following the links in the graph in a manner similar to the way the HTML web is navigated. LOD cloud's (figure 1) constantly increasing volume has a wealth of information which is of more than 14 billion RDF triples coming from a vast variety of data sources - 203 datasets. They are highly heterogeneous covering different subject domains with contribution from companies, government and public sector projects, as well as from individual Web enthusiasts. Accessing this wealth of data and making use of their full potential is still problematic. Linked data poses issues with respect to different dimensions: (a) open-world assumption of WWW data, combined with high complexity of reasoning even with OWL Lite, (b) some datasets are not suitable for reasoning, (c) publishing OWL datasets without accounting for its formal semantics. Linked data are generally unreliable as no consistency can be guaranteed. They are highly heterogeneous and hard to query. One way of accessing them is by using *reason-able views* [7] - an approach for reasoning and management of linked data. A *reason-able view (RAV)* is an assembly of independent datasets, which can be used as a single body of knowledge with respect to reasoning and query evaluation. FactForge is such a reason-able view of the web of data.

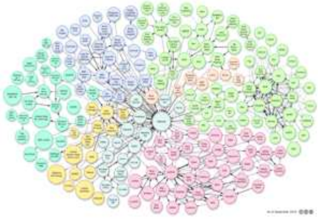


Fig. 1. Linking Open Data cloud (LOD), [9].

It gathers 8 datasets from the LOD cloud - general knowledge (DBPedia, Freebase, UMBEL, CIA World Factbook, MusicBrainz), linguistic knowledge (Wordnet, Lingvoj), geographical knowledge (Geonames). FactForge is the biggest and most heterogeneous body of factual knowledge on which inference has been performed. It comprises an overall of 1.4 billion loaded statements, 2.2 billion stored statements and 10 billion retrievable statements. FactForge is developed as an evaluation case in the European research project LarKC [8] and is used as a testbed for different large scale reasoning experiments like WebPIE [11]. It is available as a free public service at http://factforge.com**,** offering the following access facilities: (a) incremental URI auto-suggest; (b) one-node-at-a-time exploration through Forest and tabulator linked data browsers; (c) RDF Search: retrieve ranked list of URIs by keywords; (d) SPARQL end-point. One can compose SPARQL queries with predicates from multiple datasets, as shown in figure 2.

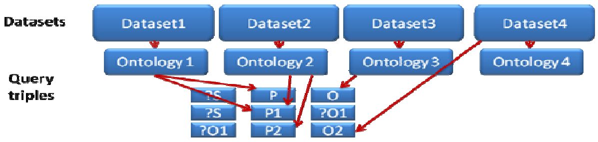


Fig. 2. SPARQL query construction for FactForge datasets.

For example, the query

```
SELECT * WHERE
          {
              ?Person dbp-ont:birthPlace ?BirthPlace ;
                      rdf:type opencyc:Entertainer ;
              ?BirthPlace geo-ont:parentFeature dbpedia:Germany .
          }
```

connects 4 datasets – DBPedia, OpenCyc, Geonames, and RDF. This powerful method to access the data from the LOD cloud has the drawback that one has to be familiar with all schemata and predicates of all datasets in FactForge in order to formulate the queries. It is even more difficult to automate the access to FactForge data and use the SPARQL end point in algorithms because of its heterogeneity. That is why we envisaged a simplified way to access the data by providing an intermediary layer - a single ontology, as shown in figure 3. To do this, we chose to align the separate schemata of FactForge with the upper-level ontology – PROTON (the Base upper-level ontology (BULO)) [14].

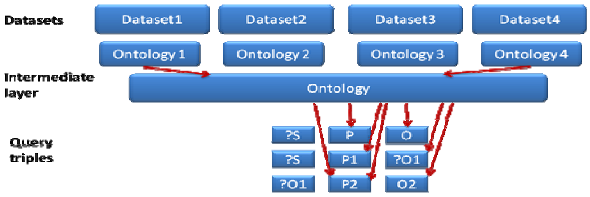


Fig. 3. SPARQL query construction for FactForge datasets in the proposed approach.

The unified access point to FactForge using a single ontology as an interface to connect to all datasets in FactForge is designed to provide an easier and simpler access to the wealth of data, higher degree of interoperability and better integration of the datasets in FactForge. It allows obtaining information from many datasets via one single ontology schema. This unified access point has important applications such as semantic search and annotation using the entities from FactForge, semantic browse and navigation, querying FactForge in natural language, and many others. It should be clear however that the upper-level ontology does not cover the full diversity of the data in the datasets. Still, for specific fine-grained queries the original data schemata and ontologies should be used.

Thus, the main objective of our project was to build a foundational ontology to explore FactForge with a balanced class hierarchy and consistent three to four levels of depth. This implied extending PROTON to obtain optimal coverage of the rich data in FactForge. In addition, the structural and conceptual differences between PROTON and the schemata organizing the datasets of FactForge like DBPedia inspired the introduction of a method for extending FactForge datasets with new instances. So, the matching model of PROTON with FactForge schemata consists in a series of iterations of enrichments at conceptual and at data levels.

2 Approaches to Matching Ontologies

Ontology matching is a key interoperability enabler for the Semantic Web, as well as a useful tactic in some classical data integration tasks. It refers to the activity of finding or discovering relationships or correspondences between entities of different ontologies or ontology modules. Matching ontologies enables the knowledge and data expressed in the matched ontologies to interoperate. Distinct methods are employed to perform ontology matching. There are syntactic and semantic matching systems [3]. In the syntactic matching the relations are computed between labels at nodes, and they are evaluated as [0, 1]. In the semantic matching the relations are computed between concepts at nodes, and they are evaluated as set theoretic relations. The semantic matching discovers semantic relationships across distinct and autonomous generic structures and recognizes relationships between matched entities, such as equivalence, subsumption, disjointness and intersection. When integrating two models, substantial difficulties may arise in transforming information from one model to the other in a heterogeneous context. Harmonising semantics is one approach for model integration by formal mapping between two domains. In this approach reference ontology is built to provide the link between the two models [3]. Except for the types of relationships that are matched between the ontologies, distinctions are made in the way the two initial ontologies are accessed. Thus, there are bidirectional and unidirectional matching methods. The bidirectional method ensures access to the two ontologies from the two ontologies, whereas the unidirectional method ensures access from one to the other ontology only [3]. Another difference in the matching methods is in the way the matching is done. There is manual and automated matching. Automated mapping is suitable for simple ontologies and simple matching tasks, where the exact accuracy of the matching is not of highest importance. In automatic matching structures that are being matched are labeled with natural language typically using WordNet. This is the vocabulary mapping. It consists in comparing Classes, Properties and Instances of two ontologies in a relation one to one. Automated matching competitions are carried out for several years now with tracks on different evaluation parameters [2], [4]. The benchmark track is run 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 best result on this track of the 2009 matching competition is F-measure of 80% [4]. Extensive surveys of automated ontology matching methods can be found in [12], [13]. The main drawback of automated ontology matching systems is that they cannot cope with ontological heterogeneity. The fact is ignored that the classes and the properties may be described in different unrelated ontologies, thus the algorithms cannot discover hidden relationships that hold between unrelated entities. Mapping by hand is considered difficult, time consuming and too long, but it derives the most accurate results. Manual mapping is suitable when maximum quality of mapping is seeked for a small quantity of concepts.

Our adopted approach is unidirectional semantic manual alignment of PROTON and the ontologies of the selected datasets of FactForge.

3 The Data

This section describes the data on which the matching in our approach is being performed, e.g. PROTON (the Base upper-level ontology (BULO) [14]) and DBPedia, Freebase and Geonames of FactForge. They are ontologies built according to different design principles. PROTON is built according to the OntoClean method [5], [6] where, for example, type and role are distinguished. It consists in evaluating the ontology concepts according to Meta properties and checking them according to predefined constraints helping to discover taxonomic errors. Using the OntoClean methodology one can discover confusions between concepts and individuals, confusions in levels of abstraction, e.g. object-level and meta-level, constraints violations, different degrees of generality.

The ontologies of FactForge datasets are made according to different methodologies. The ontologies of DBPedia and Geonames are data-driven. They provide structure and semantics to a large amount of entities in a shallow structure, but are however very different: DBPedia ontology includes many ad hoc predicates which appear in only one or several statements reflecting the variety of knowledge included in it. Geonames ontology has a concise conceptualization organized in very few well structured concepts and instances.

The upper level ontology – PROTON – is one side of the alignment process. An upper ontology is a model of the common objects that are applicable across a wide range of domains. It contains generic concepts that can serve as a domain independent foundation of other more specific ontologies. PROTON is built with a basic subsumption hierarchy comprising about 250 classes and 100 properties which provide coverage of most of the upper-level concepts necessary for semantic annotation, indexing, and retrieval.

DBPedia (http://dbpedia.org) is an RDFized version of Wikipedia. It is a collection of the structured information of Wikipedia, contained in its Infoboxes, represented in RDF and published on the Web. DBPedia ontology counts 24 first level concepts of very different degree of generality ranging from the philosophical concept of “event” through “person” and “place” to very specific concepts like “beverage”, “drug”, “protein”. Not all of DBPedia is comprised in the existing ontology. Many of the properties from the infoboxes are described separately as stand alone properties which pertain to ontological dimensions, but are not modelled in the ontology. Nevertheless some of these concepts are used in our alignment.

Freebase (http://freebase.com) is a large collaborative knowledge base, an online collection of structured data harvested from many sources, including individual wiki contribution. Freebase contains data from Wikipedia, Chemoz, NNDB, MusicBrainz and individually contributed data from its users. It has 5 million topics and no defined ontology. The entities described in this knowledge base are in structured predicate names, which reflect a hidden class hierarchy. Freebase has an overall of 19632 predicates with a structure of the predicate name in which the left most word denotes the subject domain of the property; the middle word denotes a class which is the domain of the property denoted by the last right most word, e.g.

government.legislative_session.date_ended
celebrities.romantic_relationship.end_date

Geonames (http://geonames.org) is a geographic database that covers 6 million of the most significant geographical features on Earth. It contains over 8 million geographical names and consists of 7 million unique features whereof 2.6 million populated places and 2.8 million alternate names. All features are categorized into one out of nine feature classes and further subcategorized into one out of 645 feature codes. Geonames is integrating geographical data such as names of places in various languages, elevation, population and others from various sources. All lat/long coordinates are in WGS84 (World Geodetic System 1984).

4 The Methodology

The project of building an intermediary layer between the heterogeneous data of FactForge and the end user requires matching of ontologies built according to different methods, e.g. data-driven ontologies and an upper-level ontology. This implies a translation from the one method to the other method. Further, the heterogeneity of the data in FactForge prompts the building of a unidirectional matching scheme, e.g. making FactForge accessible through PROTON predicates and entities, but not vice versa - PROTON through FactForge predicates and entities. The alignment was performed manually as the most suitable approach to find the correspondences of the small amount of upper-level concepts.

Our approach summarizes a method of matching ontologies with different methodological background – data-driven ontologies and an upper level ontology. The upper level ontology (PROTON) was chosen to be the basis for the mapping decisions, e.g. the representations of the other ontologies were translated into its model by (a) making matching rules with “ontology expressions”, (b) adding new instances with inference rules, and (c) extending the upper level ontology with classes and properties.

Thus, the adopted matching method includes:

- mapping of the concepts from PROTON to the concepts described in the datasets of FactForge, more precisely DBPedia, Freebase, Geonames
- assigning subsumption relations between entities and properties from FactForge to PROTON
- extending PROTON with classes and properties to obtain mapping at a conceptual level with FactForge
- using OWL class and property construction capabilities to represent classes and properties from FactForge and map them to PROTON classes
- extending FactForge with instances to account for the conceptual representations of the matching

The matching of the concepts and properties between DBPedia and PROTON and between Geonames and PROTON took place based on comparing the definitions of the concepts and their use. Respecting the commitment for unidirectional matching we have designed the rules with subsumption relations from FactForge to PROTON, as shown in the example below:

```
(a)   dbp:Place
              rdfs:subClassOf ptop:Location .

(b)   geo-ont:parentFeature
              rdfs:subPropertyOf ptop:subRegionOf .
```

But first, the upper level ontology PROTON was extended with new classes and properties. This was done after analyzing the content of the available data in DBPedia and Geonames with a result - a list of classes and properties which are represented within the data, and analyzing the structure of the current version of PROTON with respect to the new classes and properties. We obtained a classification of the new classes and properties using inheritance from already existing classes to the new ones. We have also used properties assigned to the new classes in order to structure them in a better way. Thus, we built a new version of PROTON with more classes and properties. Adding a new class or a new property in PROTON followed specific. A new class was added when the instances in FactForge formed a distinguishable group for which there was no concept description in PROTON. For example, DBPedia has instances for Fictional Characters, like Harry Potter, which are classified as Persons, the class FictionalCharacter was introduced in PROTON as a subclass of Person. A generic criterion for adding a new class to PROTON is the compliance with the principle of completeness of the ontology. This happens when for a given concept there are subconcepts represented in the ontology, but siblings of these concepts are missing. For example, if car and bicycle are subclasses of vehicle, but motorcycle is not, then we add motorcycle into the ontology.

To match Freebase predicates to PROTON the class construction capabilities of OWL have been used, to bind Freebase properties into classes and then match them to PROTON concepts as shown in the example (c) below:

```
(c)pfb:Location
               rdf:type owl:Restriction ;
               owl:onProperty
         <http://rdf.freebase.com/ns/type.object.type> ;
               owl:hasValue
        <http://rdf.freebase.com/ns/location.location> ;
               rdfs:subClassOf ptop:Location .
```

Here a class pfb:Location is created which is restricted to a Freebase type Location.

Another aligning method used is expression mapping. It consists in construction of classes on the basis of one of the ontologies, and mapping them to classes, or expressions of the other ontology, satisfying a relation of type many to many. For example, PROTON has a class Person and a class Profession. The subclasses of Person are Man and Woman and the subclasses of Profession are different professions, e.g. Architect, Teacher, etc. In DBPedia, Person is represented with the profession he exercises. Architect is a subclass of the class Person. Here we see a structural and conceptual difference between the PROTON model and the DBPedia model with this respect. To perform the alignment we have adapted the DBPedia model to PROTON's model in the mapping rule, as shown in figure 3.

Fig. 3. Mapping of concepts in ontologies designed according to different principles (PROTON, DBPedia).

Technically, the mapping rule looks like this:

```
(d)   dbp:Architect
             rdfs:subClassOf
                        [ rdf:type owl:Restriction ;
                                   owl:onProperty
                                           pupp:hasProfession ;
                                   owl:hasValue p-ext:Architect
                        ] .
```

The professions are modeled as instances of the class Profession in PROTON, and the single entity of DBPedia is matched to an expression in PROTON which restricts the property hasProfession to the value of the profession of interest.

The method of expression matching is not universally applicable as described above. In some cases the expressions require a reference to instances which are not included in the datasets of FactForge. This triggered the next adopted aligning method - extending the dataset of FactForge with the necessary instances, ensuring their availability to cover the entire model of the chosen basic ontology - PROTON.

FactForge is loaded into BigOWLIM, the most scalable OWL engine (http://www.ontotext.com/owlim/) supporting light-weight and high-performance reasoning with inference based on OWL Horst. BigOwlim allows the definition of custom semantics via special rules and axiomatic triples which are exploited in the process of full materialisation performed during loading. This last mechanism was used to extend FactForge with new instances by adding inference rules to the built-in ruleset. The inference rules provide the insights on what triples have to be added into the repository. They are resolved at the time of loading of the datasets into the semantic repository. For example, the inference rule (e) below:

```
(e)     p  <rdf:type> <dbp-ont:PrimeMinister>
       ------------------------------------------
```

```
       p <ptop:hasPosition> j
       j <pupp:hasTitle> <p-ext:PrimeMinister>
```

translates the DBPedia representation of someone holding a position of a Prime minister into PROTON representation. In DBPedia this is done with a type relation, whereas in PROTON this is a complex relation between a person holding a position with the title of Prime minister.

The translation of a single type relation in DBPedia can require more complexe representations, such as the ones given in example (f). Here the Freebase predicate `government.us_president` is represented as a person who holds a position in the US with the title president.

```
(f)
  a  <fb:type.object.type> <fb:government.us_president>
  ------------------------------------------------------
  a <rdf:type> <ptop:Person>
  a <ptop:hasPosition> y
  y <ptop:withinOrganization> <dbpedia:United_States>
  y <pupp:hasTitle> <p-ext:President>
```

Except for making the process of querying heterogeneous datasets easier, using one upper level ontology as an entry point to such data has another advantage. It allows to obtain information from many datasets via one single query. For example, one PROTON predicate covers three data driven predicates, e.g. PROTON `locatedIn` takes Freebase `time.event.locations`, and DBPedia `place` and `location`, as shown in the example (g) below.

```
  (g)
  dbp:place
      rdfs:subPropertyOf ptop:locatedIn .

  dbp-prop:location
      rdfs:subPropertyOf ptop:locatedIn .

  <http://rdf.freebase.com/ns/time.event.locations>
      rdfs:subPropertyOf ptop:locatedIn .
```

This makes the exploration of FactForge richer and simpler, as a query with the single PROTON predicate will retrieve information with the three other predicates from the two different datasets.

5 Results and Statistics

The outcomes of this work can be summarized as follows: (1) a new layer of unified semantic knowledge over FactForge was created by matching PROTON to FactForge schemata (2) we produced an original approach to providing similar layers to other datasets; (3) and developed a new version of PROTON ontology, which will be used in other projects. The extension of PROTON was governed by two main principles: (1) to provide coverage for the available data; and (2) to reflect the best approaches in the design of ontologies such as OntoClean methodology [5]. Table 1 shows statistics about the datasets of FactForge before and after the matching rules have been added to the semantic repository with full materialization performed. The alignment brought close to 800 million more statements and 50 million new entities available for exploration, while the matching rules cover 554 mapped classes and 103 mapped properties. The biggest number of mapped classes comes from the mapping of PROTON to Geonames' feature codes (368). As far as PROTON enrichment is concerned, 166 new classes and 73 new properties have been introduced. They cover the classes which were identified during the analysis of the instance data in FactForge and their ontologies as described in section 4.

	FactForge Initial State	FactForge with Alignments	Difference
Number of Statements	1,782,541,506	2,630,453,334	847,911,828
Number of ExplicitStatements	1,143,317,531	1,942,349,578	799,032,047
Number of Entities	354,635,159	404,798,593	50,163,434

Table 1. Statistics of FactForge

The adopted method was tested on 27 evaluation SPARQL queries selected to cover different domains, e.g. public administration, military conflicts, art and entertainment, business, medicine and to use multiple datasets from FactForge. Table 2 presents an example of an evaluation query. It is about cities around the world which have "Modigliani art works". This query is considered the ultimate test for the Semantic Web [10]. To our knowledge FactForge is the only engine capable of passing this test. The right column of the table gives the query written with PROTON predicates only. It is simpler and more intuitive than the FactForge standard one as the mapping has put all FactForge location predicates into one PROTON predicate. The number of results returned with PROTON query and with FactForge standard query are the same, presented in a slightly different way. This proves the validity of the approach.

FactForge – Standard

```
PREFIX fb: <http://rdf.freebase.com/ns/>
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbp-prop: <http://dbpedia.org/property/>
PREFIX dbp-ont: <http://dbpedia.org/ontology/>
PREFIX umbel-sc: <http://umbel.org/umbel/sc/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ot: <http://www.ontotext.com/>

SELECT DISTINCT ?painting_l ?owner_l ?city_fb_con ?city_db_loc
?city_db_cit

WHERE {
          ?p fb:visual_art.artwork.artist
                                  dbpedia:Amedeo_Modigliani ;
             fb:visual_art.artwork.owners [
                fb:visual_art.artwork_owner_relationship.owner
                                                      ?ow ] ;
             ot:preferredLabel ?painting_l.
          ?ow ot:preferredLabel ?owner_l .
```

FactForge - PROTON

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ot: <http://www.ontotext.com/>
PREFIX ptop: <http://proton.semanticweb.org/protont#>
PREFIX ploc: <http://proton.semanticweb.org/protonl#>
PREFIX p-ext: <http://proton.semanticweb.org/protonue#>

SELECT DISTINCT ?painting ?owner ?city

WHERE {
          ?p p-ext:author dbpedia:Amedeo_Modigliani ;
               p-ext:ownership [ ptop:isOwnedBy ?ow ] ;
               ot:preferredLabel ?painting .
          ?ow ot:preferredLabel ?owner .
          ?ow ptop:locatedIn [  rdf:type ploc:City ;
                                ot:preferredLabel ?city].
      }
```

```
       OPTIONAL { ?ow fb:location.location.containedby
                  [ ot:preferredLabel ?city_fb_con ] }
        OPTIONAL { ?ow dbp-prop:location ?loc.
                    ?loc rdf:type umbel-sc:City ;
                       ot:preferredLabel ?city_db_loc }
        OPTIONAL { ?ow dbp-ont:city [ ot:preferredLabel
                                       ?city_db_cit ] }
}
```

Table 2. Modigliani Test Query

In cases where several FactForge predicates are matched to a single PROTON predicate, like the location predicates mentioned earlier in the paper, the PROTON queries return more results than FactForge – Standard queries. Thus, the advantages of the approach to have a single access point to the Linked Open Data (LOD) cloud are twofold: they provide access by simpler queries and they provide leveraged query results.

5 Future work

We envision in the future building a two level intermediary layer to access FactForge and then LOD cloud mapping PROTON to UMBEL (http://www.umbel.org/documentation.html) – “a lightweight subject concept reference structure for the Web” with about 20 000 subject concepts based on OpenCyc (http://www.cyc.com/opencyc/). We intend to cover more datasets from the LOD cloud, and to experiment with the balance between the data from the LOD and FactForge datasets and the ontological schemata describing them.

References

1. Christian Bizer, Tom Heath, Tim Berners-Lee. Linked Data – The Story so Far. In: Heath, T., Hepp, M. and Bizer, C. (eds.) Special Issue on Linked Data, International Journal on Semantic Web and Information Systems (IJSWIS), http://linkeddata.org/docs/ijswis-special-issue, (2009).
2. Caterina Caracciolo, Jérôme Euzenat, Laura Hollink, Ryutaro Ichise, Antoine Isaac, Véronique Malaisé, Christian Meilicke, Juan Pane, Pavel Shvaiko, Heiner Stuckenschmidt, Ondˇrej Šváb-Zamazal, and Vojtˇech Sváte, Results of the Ontology Alignment Evaluation Initiative 2008, (2008).
3. Mohamed El-Mekawy and Anders Östman. Semantic Mapping: an Ontology Engineering Method for Integrating Building Models in IFC and CITYGML. In Proceedings of the 3rd ISDE Digital Earth Summit 12-14 June, 2010, Nessebar, Bulgaria, (2010).
4. J. Euzenat, A. Ferrara, L. Hollink, A. Isaac, C. Joslyn, V. Malaisé, C. Meilicke, A. Nikolov, J. Pane, M. Sabou, F. Scharffe, P. Shvaiko, V. Spiliopoulos, H. Stuckenschmidt, O. Šváb-Zamazal, V. Svátek, C. Trojahn dos Santos, G. Vouros, S. Wang: Results of the Ontology Alignment Evaluation Initiative 2009. In Proceedings of the 4th Ontology Matching Workshop at ISWC, (2009).
5. Nicola Guarino and Christopher Welty. “Evaluating Ontological Decisions with OntoClean.” Communications of the ACM, 45(2): 61-65 (2002).

6. Michael Grüninger and Mark Fox. Methodology for the Design and Evaluation of Ontologies. Proceedings of IJCAI95's Workshop on Basic Ontological Issues in Knowledge Sharing (1995).
7. Atanas Kiryakov, Damyan Ognyanoff, Ruslan Velkov, Zdravko Tashev, Ivan Peikov. LDSR: Materialized Reason-able View to the Web of Linked Data. In: Proceedings of OWLED 2009. Chantilly, USA, 23-24 October 2009 (2009).
8. LarKC project. The Large Knowledge Collider - http://www.larkc.eu/, (2010).
9. Linking Open Cloud Diagramme, as of September 2010, http://lod-cloud.net/.
10. Richard MacManus. The Modigliani Test: The Semantic Web's Tipping Point. http://www.readwriteweb.com/archives/the_modigliani_test_semantic_web_tipping_point.php (2010).
11. Vassil Momchev, Matthias Assel, Alexey Cheptsov, Barry Bishop, Luka Bradesko, Christoph Fuchs, Georgina Gallizo, Spyros Kotoulas, Gaston Tagni D5.5.3 Report on platform validation and recommendation for next version, LarKC EU-IST-2008-215535, (2010).
12. P. Shvaiko, J. Euzenat: Ten Challenges for Ontology Matching. In Proceedings of ODBASE, (2008).
13. P. Shvaiko, J. Euzenat: A Survey of Schema-based Matching Approaches. Journal on Data Semantics, 2005.
14. Ivan Terziev, Atanas Kiryakov, Dimitar Manov. „D.1.8.1 Base upper-level ontology (BULO) Guidance" Deliverable of EU-IST Project IST – 2003 – 506826 SEKT (2005).

Ontology Alignment in the Cloud

Jürgen Bock, Alexander Lenk, and Carsten Dänschel

FZI Research Center for Information Technology, Karlsruhe, Germany
`{bock,lenk,daenschel}@fzi.de`

Abstract. The problem of ontology alignment is prominent for applications that operate on integrated semantic data. With ontologies becoming numerous and increasingly large in size, scalability is an important issue for alignment tools. This work introduces a novel approach for computing ontology alignments using cloud infrastructures. An alignment algorithm based on particle swarm optimisation is deployed on a cloud infrastructure, taking advantage of its ability to harness parallel computation resources. The deployment is done with a focus on parallel efficiency, taking into account both communication latency and computational inhomogeneity among parallel execution units. Complementing previous experiments showing the effectiveness of the alignment algorithm, this paper contributes an experiment executed "in the cloud", which demonstrates the scalability of the approach by aligning two large ontologies from the biomedical domain.

1 Introduction

Ontology alignment is a problem prominent in semantic applications, the semantic web, and the linked data web. It is based on the observation that ontologies[1] are heterogeneous models, often representing a similar or equal domain of interest, and hence have a certain overlap. Accordingly an ontology alignment is defined as a set of correspondences between ontological entities, *i.e.* classes, properties, and individuals, of two ontologies.

Two examples of where ontology alignment plays an important role are the linked data web [2] and the biomedical domain [21]. These examples also demonstrate the necessity for alignment tools to be able to deal with very large ontologies, which currently constitutes a problem for most alignment algorithms. Furthermore, it can be observed that ontologies evolve gradually, *i.e.* changes typically occur by adding / removing / modifying single entities while the largest part of the ontology remains unchanged. Thus, due to the typically slow changes, alignments do not need to be recomputed completely each time, but rather incrementally maintained and adjusted according to the evolving ontologies. Furthermore, in typical information systems that utilise alignments, the refresh period

---

[1] There is some disagreement among semantic web and linked data web communities about whether data sources in the linked data web can be called ontologies. In this paper the term *ontology* is used to cover all types of semantic data sources.

for alignments is not time-critical, *i.e.* alignments do not need to be recomputed ad-hoc, but can be adjusted offline, *e.g.* in nightly alignment jobs.

This paper addresses the scalability problem of ontology alignment by utilising cloud computing as a massively parallel computation infrastructure. The algorithm to be deployed in the cloud is an alignment algorithm based on particle swarm optimisation (PSO) [4]. A PSO-based approach to the ontology alignment problem has several characteristics that meet the aforementioned observations:

1. The algorithm provides a meta-heuristic, which is independent of the objective function to be optimised. Hence it is straightforward to exchange or adapt the objective function according to the alignment scenario at hand.
2. The algorithm works incrementally, which has two beneficial effects: Firstly, the algorithm can be interrupted at any time providing the best alignment that has been discovered so far. Secondly, the algorithm can be provided with an initial (partial) alignment as a start configuration to be refined. This in particular allows for alignment evolution of gradually changing ontologies.
3. The algorithm is inherently parallelisable allowing for efficient execution on parallel computing infrastructures.

The first two issues are inherent to the application of the PSO meta-heuristic. However, the usability of the approach depends on an efficient deployment on parallel computing infrastructures. Computation and adjustment of alignments on an irregular basis usually does not justify the acquisition of large parallel hardware infrastructures.

Cloud computing infrastructures are scalable and can be used for data intensive parallelisable jobs, as it has successfully been shown *e.g.* in the field of bio-informatics [20]. Recently, cloud computing has also been recognised as a promising technique to provide scalability for web data processing [18]. The use of cloud infrastructures provides the possibility to access a large number of computing resources in a convenient way without the need of operating one's own, expensive data centre. Many offerings by cloud providers such as Amazon Web Services$^{\mathrm{TM}}$ (AWS)[2] are based on a pay-per-use pricing model with an hourly rate. This allows for flexible use of computation resources, since accessing large amounts of computation power for a short period of time comes at the same costs as using just a few resources for a long period of time.

The remainder of this paper is structured as follows. In Sect. 2 relevant background knowledge and related work is presented. Section 3 introduces the approach of ontology alignment by PSO and demonstrates its parallel scalability. Justified by these insights, Sect. 4 discusses the design and implementation of a cloud-based deployment of the formerly introduced approach. Experimental results are presented in Sect. 5 demonstrating both the effectiveness by referring to previous benchmarks, and the scalability by using a real-world biomedical alignment scenario. Section 6 summarises the contributions and provides an outlook on future work.

---

[2] `http://aws.amazon.com`

2 Foundations

This section introduces ontology alignment, particle swarm optimisation (PSO), and cloud computing in more details.

2.1 Ontology Alignment

An ontology alignment is defined as a set of correspondences between ontological entities [10], *i.e.* classes, properties, and individuals, of two ontologies. In this approach, an entity can correspond to at most one entity of the other ontology, which denotes what Euzenat and Shvaiko call an ?:? alignment [10]. Ontology alignment detection can be perceived as an optimisation problem [4], where the task is to find the optimal alignment of two ontologies w.r.t. the criteria denoted in terms of an evaluation function. These criteria can be based on several basic matching techniques [10, Chap. 4] or other measures, respecting domain specific and modelling language specific characteristics of the ontologies to be aligned.

A representative overview of the state-of-the-art in ontology alignment is given by Euzenat and Shvaiko [10, Chap. 6] and by the yearly Ontology Alignment Evaluation Initiative (OAEI) [9]. Most systems focus on the improvement of alignment quality, whereas the scalability problem has attracted interest only recently. As the OAEI participation shows, only few systems are capable of dealing with large ontologies [9, Sect. 10].

2.2 Particle Swarm Optimisation (PSO)

PSO is a biologically and socioculturally-inspired meta-heuristic [12, 13, 8], originally proposed in 1995 by Kennedy and Eberhart [12]. It has become a major research field since then.

PSO algorithms use a population of particles to find the optimal parameter configuration with respect to one or more objective functions. Each particle represents a candidate solution. A particle's *fitness* is evaluated using objective function(s). During initialisation, each particle in the swarm is assigned a random position in the parameter space. A PSO algorithm runs iteratively, where in each iteration, each particle adjusts its position in the parameter space by adding a *velocity* vector to its current position. These particle updates can be performed in parallel. The velocity vector for a particle is determined by the particle's previous velocity (*inertia*), the best position that has been visited by any neighbour of the particle so far (*social component*), and the best position, the particle itself has visited so far (*cognitive component*). Regarding the social component, several models of defining particle neighbourhoods have been analysed. Such models are called social network structures or *topologies*. A PSO algorithm is called *gBest* (global best) PSO, if the neighbourhood of a particle consists of the whole swarm. Thus the *gBest* PSO is a special case of the *lBest* (local best) PSO, where the neighbourhood of a particle comprises only a subset of the swarm [8, Chap. 12].

It is important to note, that the particle swarm search heuristic works without any knowledge about the objective function(s). Hence an objective function is replaceable and adjustable to arbitrary optimisation problems.

2.3 Cloud Computing

Cloud computing is a new paradigm that has been evolving over the last few years. Most definitions of cloud computing have in common that cloud computing offerings can be categorised using the “Everything as a Service” (XaaS) model. A more detailed view of this model is the “Cloud Computing Stack” [16]. According to the XaaS model the three main service classes are “Software as a Service” (SaaS), “Platform as a Service” (PaaS), and “Infrastructure as a Service” (IaaS). While SaaS offerings usually provide an interface directly to the end user by providing a Web service interface or a graphical user interface (GUI) the PaaS and IaaS offerings can be used by software architects to build new SaaS services on top of them. PaaS offerings usually provide a platform where the software developer can deploy the new services [16].

The IaaS offerings at the “Basic Infrastructure Services” level, give the developer full control over the servers that are running his software. At this level the user can deploy new machines using Web service technologies. This offers the power and flexibility to work on a machine level without running one’s own data center and so having convenient access to new resources. Offerings such as Amazon EC2 give users the opportunity to automatically deploy hundreds of virtual machines within minutes. Thus, it is possible to build highly scalable applications that can scale up and down in short periods. One famous example of a successful cloud offering is Animoto [1]. The Animoto application transforms music files and photos into small video slide shows. After offering their service to users on a social network the demand of virtual machines went from about 40 to 3500 [15]. This scalability was only possible by designing the Animoto software as a distributed algorithm deployed on Amazon EC2.

Another interesting feature of cloud offerings is the typical pay-as-you-go pricing model. This means that users only pay for the resources they are really using. Having such a pricing model it makes no difference if one single server is running for 10 hours or if 10 servers are running for just one hour. The New York Times used this pricing scheme when they built their “TimesMaschine”. By using Amazon EC2 they were able to convert their whole archive (4 TB of scanned TIFF files), spanning the years 1851-1922, to web documents within 24 hours and total costs of US$ 890 [11].

In the context of semantic technologies and the semantic web, cloud computing technologies have been successfully applied, mainly for RDF storage [22, 19, 18], querying [18], and materialisation of RDF/OWL knowledge bases [24, 23].

3 Ontology Alignment by Particle Swarm Optimisation

Considering ontology alignment as an optimisation problem, a novel discrete PSO algorithm (DPSO) has been developed to find an optimal alignment of two ontologies [4]. The algorithm has been implemented under the name MapPSO[3] and is based on a DPSO algorithm by Correa *et al.* [6], which implements efficient attribute selection in data mining classification tasks.

---

[3] `http://mappso.sourceforge.net`

3.1 Algorithm

In the case of ontology alignment, each particle represents a valid candidate alignment. With a *swarm* of particles being initialised randomly, in each iteration, each particle evaluates the alignment it represents via an objective function comprising various basic matching techniques [10, Chap. 4] as well as the size of an alignment. Typically there is only a partial overlap of ontologies, so the alignment algorithm strives for finding the largest alignment, *i.e.* maximum number of correspondences, of high quality. *Base matchers* can evaluate a correspondence according to different characteristics, such as lexical or linguistic similarity of entity labels or comments. Moreover, correspondences do not necessarily have to be evaluated in isolation, *e.g.* a correspondence of two classes can get a better evaluation if a correspondence of their superclasses is also part of the alignment represented by this particle. As discussed in Sect. 2.2 the objective function and thus the base matchers are replaceable and hence can be adapted to particular alignment scenarios. For instance in biomedical ontologies specific OBO[4] [21] annotations can be exploited to evaluate the similarity between entities.

Similar to the previous work by Correa *et al.* [6] and the original binary PSO algorithm by Kennedy and Eberhart [14], particle movements are determined by *proportional likelihoods*, since the traditional notion of velocity vectors is not applicable for binary search spaces, as it is the case for ontology alignment. The likelihood for a correspondence to be contained in a particle in the subsequent iteration is increased, if the correspondence exists in the global best (social component) or personal best (cognitive component) particle configuration.

This update mechanism allows for a guided convergence of the swarm to a global optimum, which represents the best alignment with respect to the basic matching techniques specified as objective function. Moreover, the size of each particle varies during the execution of the algorithm influenced by the size of the global best alignment. This causes the algorithm to search for the optimum size of the alignment, *i.e.* partial overlap, as well.

3.2 Parallel Efficiency

A detailed theoretical discussion of the correlation between population size and number of iterations with respect to the swarm convergence goes beyond the scope of this paper. Figure 1 illustrates an empirical analysis of the convergence towards the optimum[5] when aligning small ontologies of the OAEI benchmark dataset. The figure shows clearly that a larger number of particles results in faster convergence by reducing the number of required iterations and thus reducing wall-clock runtime. Where it takes about 225 iterations for a swarm of 8 particles to reach an alignment of high quality, the same result is achieved in only 90 iterations by a swarm of 64 particles. Using only 2 particles does not reach an equally good result within 300 iterations.

---

[4] Open Biomedical Ontologies

[5] Note that the optimum depends on the chosen base matchers and thus not necessarily has to be 0.

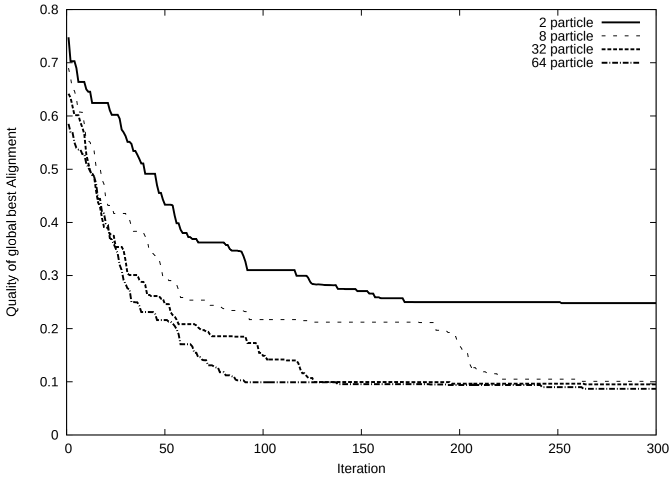


Fig. 1. Convergence of the PSO algorithm for different population sizes using test case *101* of the OAEI benchmarks. Alignment quality is represented as the fitness of the global best particle, where a smaller value represents better alignment quality.

4 Deployment in the Cloud

To obtain the required scalability, the PSO-based alignment algorithm has been ported to Amazon Web Services$^{\text{TM}}$ (AWS), the IaaS cloud service of Amazon®. AWS is one of the biggest IaaS providers with a large and active community, and is representative for any other IaaS provider in the context of this work.

The deployment has been realised using a server-worker pattern. Several *workers* evaluate and update several local particles each, while they are managed by a central *server*. Each worker determines the local best alignment among the results of its particles and sends it to the server. The server determines the global best alignment, then broadcasts it and synchronises the workers. Exchange of information between server and workers is realised by the Amazon Simple Queue Service (SQS)[6] or via TCP/IP. The exchange of concrete particle states is realised by the Amazon Simple Storage Service (S3)[7] for reasons of scalability, reliability, and parallel access.

4.1 Challenges

Deploying the algorithm on virtual machines connected by a network bears two main challenges. Firstly, the communication latency is much higher when communicating via network than via main memory. Hence finding and broadcasting the global best particle leads to a higher communication overhead, which slows down the algorithm and creates unwarranted costs.

---

[6] `https://aws.amazon.com/sqs/`

[7] `https://s3.amazonaws.com/`

Secondly, the computation times of workers in a particular iteration differ. This difference occurs mainly for two reasons: The unpredictable performance of the virtual environment, and the varying particle sizes. The performance of the virtual machine depends on its mapping to real hardware. For example a "noisy" neighbour, *i.e.* another program sharing the same real hardware via a different virtual machine, can slow down network communication. In turn, a virtual machine can utilise more computing power if there are no or only inactive neighbours. This results in unbalanced performance of the virtual machines. The random initialisation of particles and their different sizes add to this discrepancy in computation time. A small particle needs less computation time than a big particle, therefore a worker with smaller particles will require less runtime per iteration. The random initialisation of particles with different sizes is necessary to search for the optimal alignment size and thus be able to identify *partial* overlaps of ontologies. This computation time discrepancy causes fast workers to idle while waiting for the slower workers and thus decreases parallel efficiency.

4.2 Increasing Parallel Efficiency

The challenges of deploying the algorithm to a cloud-based infrastructre identified have been addressed and solutions are proposed as follows.

Addressing Latency. Amazon SQS was used as a means for communication between server and workers. Amazon advertises SQS as reliable, scalable, and simple. However, it turned out that SQS has high latency. For reducing the network latency, direct communication has been implemented using the TCP/IP protocol. Apart from reduced latency, this also resulted in a reduction of the additional communication overhead caused by multiple workers.

The latency is reduced further by only sending particle states when necessary, *i.e.* when a better particle state has been found. To achieve this a worker sends the fitness value of its local best particle to the server and writes the particle state itself into the S3 database. The server then broadcasts the global best fitness and the according database location to all worker instances.

The number of read operations is minimal, since in the PSO topology used, each particle must know about the global best. In case the global best does not change, no worker has to read a particle state. The (unlikely) worst case in terms of communication latency happens if every worker finds a new best alignment and thus writes its particle state before it has to read a new one found by another worker in the same iteration.

Addressing Runtime Discrepancy. The effect that workers require different runtimes for particle fitness computation can be minimised by *particle pooling*, *i.e.* having each worker instance computing multiple particles. Using few particles per worker results in high runtime variance between workers caused by different particle sizes and the resulting uneven workload. Using more particles per worker, *i.e.* the workload for each worker being the sum of workloads contributed by each of its particles, averages the runtime because a rather uniform distribution of small and big particles per worker is expected. Using a multi-particle approach also increases parallel efficiency due to the increased overall runtime required to evaluate and update several particles. By increasing the runtime the proportion of time used for communication decreases and thus parallel efficiency is increased.

Using *asynchronous* particle updates is another way to compensate the runtime discrepancy. When using synchronous particle updates every worker has to wait for the slowest worker and thus is wasting computation time. In the asynchronous communication mode workers keep on evaluating and updating their particles until they find a particle state that is better than the global best they know about. They send this particle state to the server and continue. The server broadcasts the new particle state if it is better than the global best known by the server. Preventing workers to idle drastically increases parallel efficiency.

Introducing an asynchronous particle update strategy has an effect on the underlying particle swarm meta-heuristic. Having not all particles exchanging information at the same time step has a similar effect than changing the social network structure of the PSO from a star topology to a cluster topology[8] [8, Chap. 12], which results in fast communication between particles on the same worker compared to the communication between workers themselves. A clustered (*lBest*) topology in general results in slower convergence but better robustness compared to the star (*gBest*) topology [8, Chap. 12].

Furthermore, a loss of information can be observed compared to the synchronous update mechanism. This is due to the fact, that particles compute iterations with possibly outdated information about the global best, which might be available on a worker that is still computing another particle. This problem has been analysed by Lewis *et al.* [17]. Their investigations revealed that the information loss can be compensated by the increased number of computations that are possible due to the reduced waiting time.

The beneficial effect on the runtime for an alignment is reflected in Fig. 2, showing runtime behaviour for two medium sized ontologies for both synchronous and asynchronous particle updates. As expected, using an asynchronous communication mode does not have any effect if only a single worker is used. However, for 16 workers that need to communicate their local best results in each iteration, a clear runtime improvement of almost 50 % can be observed.

One further effect resulting from the asynchronous particle updates is that workers hosting mainly small particles can complete an iteration more quickly than those hosting mainly large particles. Therefore small particles tend to compute more iterations and thus influence the swarm stronger than it would be the case in the synchronous mode. This is beneficial to the overall runtime of the algorithm, because the average size of particles is smaller and therefore iterations are faster.

---

[8] While in a star topology, every particle shares information with every other particle, the cluster topology allows only groups of particles to communicate with each other, while the groups themselves exchange information only via dedicated particles, which are part of two clusters.

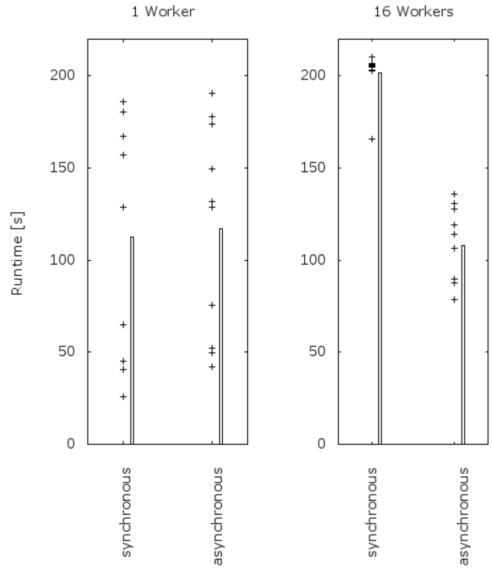


Fig. 2. Synchronous vs. asynchronous particle updates using 1 and 16 workers with 1 particle per worker. Bars denote an average of 10 runs (individual runs denoted left of the bars). Depicted is the total runtime for an alignment of the mouse ontology from the OAEI anatomy track with itself.

5 Evaluation

Since for large ontologies there are no reference alignments available, effectiveness and scalability of the approach need to be evaluated separately. While it can be shown for small benchmark ontologies, that the algorithm produces correct results using a problem specific parameter configuration, a separate experiment has been conducted evaluating scalability using a cloud infrastructure.

5.1 Effectiveness

Previous evaluations of the MapPSO system at the Ontology Alignment Evaluation Initiative (OAEI) campaigns 2008 and 2009 [3, 5] have shown the effectiveness of the approach. In particular with respect to relaxed precision and recall metrics [7] the OAEI 2009 organisers report that "[...] 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 systems." [9]. The reason for this difference to classical metrics is most likely the rather generic parameter configuration that has been used for all benchmark tests, without adaptation to the single alignment scenarios. This generic parameter configuration is a restriction that is of minor importance in real-world alignment scenarios.

5.2 Scalability

In order to demonstrate the scalability of this approach, an experiment was conducted aligning two large ontologies from the biomedical domain. The chosen ontologies were the Gene Ontology (GO)[9] with 31,650 classes, and the Medical Subject Headings (MESH) Ontology[10] with 15,343 classes, both converted to the OWL format. Specific base matchers were used that take advantage of the class annotations resulting from the conversion from OBO to OWL.

In the experiment a population of 128 particles was used distributed on 16 Amazon EC2 instances (workers) and thus utilising particle pooling by computing 8 particles per worker. The EC2 instances were of the type "High-CPU Extra Large Instance, 7 GB of memory, 20 EC2 Compute Units (8 virtual cores with 2.5 EC2 Compute Units each), [. . .] 64-bit platform, [where] one EC2 Compute Unit (ECU) provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor"[11]. The setup was chosen in a way, that the number of particles on each instance matches the number of virtual cores, thus enabling efficient multi-threading of particle computation.

The experiment was run for 3.5 h, where due to the asynchronous particle updates, workers computed between 50 and 75 iterations. The difference in the number of iterations illustrates the different computational efforts of each worker. Workers that achieved a lower number of iterations than other workers at the same time, are most likely computing smaller particles on average. This illustrates the runtime discrepancy addressed in Sect. 4.1 and the increased parallel efficiency gained by using an asynchronous approach. By using a synchronous approach the slowest particle constitutes an upper bound and thus a maximum of 50 iterations could have been computed in 3,5 h. The asynchronous approach, however, computes an average of 62.875 iterations, which is an increase of 25 %.

Since there is no reference alignment available for GO and MESH as there is none for any other large ontologies, no statement about the result quality can be made. The experiment, however, demonstrates the scalability of the proposed approach, operating on large ontologies and converging to an alignment.

6 Conclusion

The problem of large-scale ontology alignment detection has been addressed by a PSO-based approach. PSO has the characteristics of (*i*) being independent of its objective function(s) and thus easily adaptable to various ontology alignment scenarios, (*ii*) being incremental, *i.e.* able to refine an alignment when input ontologies evolve, and (*iii*) being inherently parallelisable. The latter aspect has been exploited in this paper by deploying the algorithm in the AWS cloud. This deployment is making use of the emerging cloud computing paradigm, which

---

[9] `http://www.geneontology.org/`

[10] `http://www.nlm.nih.gov/mesh/meshhome.html`

[11] `http://aws.amazon.com/ec2/#instance`, accessed 2010/06/19

provides an on-demand infrastructure for dynamic computational needs, as it is the case for alignment refinements of gradually evolving ontologies.

When utilising parallel computation infrastructures on a pay-per-use basis, such as cloud offerings, it is crucial to maximise parallel efficiency. Due to the variable particle sizes used in this approach, different computation times for each particle in each iteration can be observed. Thus particle pooling, *i.e.* multiple particles per cloud instance, as well as asynchronous particle updates have been introduced to the algorithm in order to reduce the time wasted by idling particles.

Previous experiments in the course of OAEI participations have shown the effectiveness of the PSO-based approach. Complementing these results, the scalability via parallelisation has been shown for two large biomedical ontologies.

Having realised the successful deployment of an ontology alignment algorithm in the cloud using an IaaS provider, it is a straightforward extension to provide ontology alignment itself as a web service (SaaS) based on a dynamic and scalable infrastructure. This enables large-scale ontology alignment for every user without bothering about hardware requirements. Together with the anytime behaviour of the PSO-based algorithm, business models comprising the dimensions computation time, price, and alignment quality can be created.

Acknowledgement

The presented research was partially funded by the German Federal Ministry of Economics (BMWi) under the project Theseus (number 01MQ07019).

References

1. Animoto Productions: animoto - the end of slideshows:. online, `http://www.animoto.com`, last seen: 2010/06/16
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
3. Bock, J., Hettenhausen, J.: MapPSO Results for OAEI 2008. In: Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008). vol. 431. CEUR Workshop Proceedings, http://ceur-ws.org (2008)
4. Bock, J., Hettenhausen, J.: Discrete Particle Swarm Optimisation for Ontology Alignment. Information Sciences (Special Issue on Swarm Intellligence and Applications) (2010), article in press
5. Bock, J., Liu, P., Hettenhausen, J.: MapPSO Results for OAEI 2009. In: Proceedings of the 4th International Workshop on Ontology Matching (OM-2009). vol. 551, pp. 193–199. CEUR Workshop Proceedings, http://ceur-ws.org (2009)
6. 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). pp. 35–42. ACM (2006)
7. Ehrig, M., Euzenat, J.: Relaxed Precision and Recall for Ontology Matching. In: Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies. vol. 156, pp. 25–32. CEUR Workshop Proceedings, http://ceur-ws.org (2005)

8. Engelbrecht, A.P.: Fundamentals of Computational Swarm Intelligence. Wiley (2007)
9. Euzenat, J., Ferrara, A., Hollink, L., Isaac, A., Joslyn, C., Malaisé, V., Meilicke, C., Nikolov, A., Pane, J., Sabou, M., Scharffe, F., Shvaiko, P., Spiliopoulos, V., Stuckenschmidt, H., Šváb-Zamazal, O., Svátek, V., dos Santos, C.T., Vouros, G., Wang, S.: Results of the Ontology Alignment Evaluation Initiative 2009. In: Proceedings of the 4th International Workshop on Ontology Matching (OM-2009). vol. 551. CEUR Workshop Proceedings, http://ceur-ws.org (2009)
10. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer (2007)
11. Hilley, D.: Cloud Computing: A Taxonomy of Platform and Infrastructure-level Offerings. Tech. rep., College of Computing, Georgia Institute of Technology (2009)
12. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks. vol. 4, pp. 1942–1948. IEEE Computer Society (1995)
13. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann (2001)
14. Kennedy, J., Eberhart, R.C.: A Discrete Binary Version of the Particle Swarm Algorithm. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. vol. 5, pp. 4104–4108. IEEE Computer Society (1997)
15. Killalea, T.: Building Scalable Web Services. Queue 6(6), 10–13 (2008)
16. Lenk, A., Klems, M., Nimis, J., Tai, S., Sandholm, T.: What's inside the Cloud? An architectural map of the Cloud landscape. In: Proceedings of the ICSE Workshop on Software Engineering Challenges of Cloud Computing (CLOUD). pp. 23–31. IEEE Computer Society (2009)
17. Lewis, A., Mostaghim, S., Scriven, I.: Asynchronous Multi-Objective Optimisation in Unreliable Distributed Environments, Studies in Computational Intelligence, vol. 210, pp. 51–78. Springer (2009)
18. Mika, P., Tummarello, G.: Web Semantics in the Clouds. IEEE Intelligent Systems 23(5), 82–87 (2008)
19. Newman, A., Li, Y.F., Hunter, J.: Scalable Semantics – the Silver Lining of Cloud Computing. In: Proceedings of the IEEE Fourth International Conference on eScience. pp. 111–118. IEEE Computer Society (2008)
20. Qiu, X., Ekanayake, J., Beason, S., Gunarathne, T., Fox, G., Barga, R., Gannon, D.: Cloud Technologies for Bioinformatics Applications. In: Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS). pp. 1–10. ACM (2009)
21. Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L.J., Eilbeck, K., Ireland, A., Mungall, C.J., Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.A., Scheuermann, R.H., Shah, N., Whetzel, P.L., Lewis, S.: The OBO Foundry: Coordinated Evolution of Ontologies to Support Biomedical Data Integration. Nature Biotechnology 25(11), 1251–1255 (2007)
22. Stein, R., Zacharias, V.: RDF On Cloud Number Nine. In: Proceedings of the 4th Workshop on New Forms of Reasoning for the Semantic Web: Scalable & Dynamic. pp. 11–23. CEUR Workshop Proceedings, http://ceur-ws.org (2010)
23. Urbani, J., Kotoulas, S., Maassen, J., van Harmelen, F., Bal, H.: OWL Reasoning with WebPIE: Calculating the Closure of 100 Billion Triples. In: Proceedings of the 7th Extended Semantic Web Conference (ESWC). LNCS, vol. 6088, pp. 213–227. Springer (2010)
24. Urbani, J., Kotoulas, S., Oren, E., van Harmelen, F.: Scalable Distributed Reasoning using MapReduce. In: Proceedings of the 8th International Semantic Web Conference (ISWC). LNCS, vol. 5823, pp. 634–649. Springer (2009)

Results of the Ontology Alignment Evaluation Initiative 2010*

Jérôme Euzenat[1], Alfio Ferrara[2], Christian Meilicke[3], Andriy Nikolov[4], Juan Pane[5],
François Scharffe[1], Pavel Shvaiko[6], Heiner Stuckenschmidt[3], Ondřej Šváb-Zamazal[7],
Vojtěch Svátek[7], and Cássia Trojahn[1]

[1] INRIA & LIG, Montbonnot, France
`{jerome.euzenat,francois.scharffe,cassia.trojahn}@inrialpes.fr`
[2] Universita degli studi di Milano, Italy
`ferrara@dico.unimi.it`
[3] University of Mannheim, Mannheim, Germany
`{christian,heiner}@informatik.uni-mannheim.de`
[4] The Open University, Milton Keynes, UK
`A.Nikolov@open.ac.uk`
[5] University of Trento, Povo, Trento, Italy
`pane@dit.unitn.it`
[6] TasLab, Informatica Trentina, Trento, Italy
`pavel.shvaiko@infotn.it`
[7] University of Economics, Prague, Czech Republic
`{ondrej.zamazal,svatek}@vse.cz`

Abstract. Ontology matching consists of finding correspondences between entities of two ontologies. OAEI campaigns aim at comparing ontology matching systems on precisely defined test cases. Test cases can use ontologies of different nature (from simple directories to expressive OWL ontologies) and use different modalities, e.g., blind evaluation, open evaluation, consensus. OAEI-2010 builds over previous campaigns by having 4 tracks with 6 test cases followed by 15 participants. This year, the OAEI campaign introduces a new evaluation modality in association with the SEALS project. A subset of OAEI test cases is included in this new modality which provides more automation to the evaluation and more direct feedback to the participants. This paper is an overall presentation of the OAEI 2010 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 systems [9]. 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 improve their systems.

---

[*] This paper improves on the "First results" initially published in the on-site proceedings of the ISWC workshop on Ontology Matching (OM-2010). The only official results of the campaign, however, are on the OAEI web site.

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

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) [18]. Then, unique OAEI campaign occurred in 2005 at the workshop on Integrating Ontologies held in conjunction with the International Conference on Knowledge Capture (K-Cap) [1]. Starting from 2006 through 2009 the OAEI campaigns were held at the Ontology Matching workshops collocated with ISWC [8; 7; 3; 6]. Finally in 2010, the OAEI results were presented again at the Ontology Matching workshop collocated with ISWC, in Shanghai, China[2].

The main novelty of this year is the adoption of an environment for automatically processing evaluations (§2.2), which has been developed in coordination with the SEALS project[3]. This project aims at providing standardized datasets, a software infrastructure for automatically executing evaluations, and evaluation campaigns for typical semantic web tools, including ontology matching. This year, a subset of OAEI datasets is included in the SEALS modality. The goal is to provide better direct feedback to the participants and a more common ground to the evaluation.

We have discontinued the oriented alignment track of the last year because there was not enough organizational resources to guarantee a satisfying evaluation.

This paper serves as an introduction to the evaluation campaign of 2010 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 evaluation methodology that has been used. Sections 3-7 discuss in turn the settings and the results of each of the test cases. Section 8 overviews lessons learned from the campaign. Finally, Section 9 outlines future plans and Section 10 concludes the paper.

2 General methodology

We first present the test cases proposed this year to OAEI participants (§2.1). Then, we present the evaluation environment, which has been used by participants to test their systems and launch their evaluation experiments for the campaign (§2.2). Next, we describe the steps of the OAEI campaign (§2.3-2.6) and report on the general execution of the campaign (§2.7).

2.1 Tracks and test cases

This year's campaign has consisted of 4 tracks gathering 6 data sets and different evaluation modalities:

The benchmark track (§3): Like in previous campaigns, a systematic benchmark series have been proposed. The goal of this benchmark series is to identify the areas in which each alignment 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.

---

[2] `http://om2010.ontologymatching.org`

[3] `http://www.seals-project.eu`

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):** The goal of this track is to find all correct correspondences within a collection of ontologies describing the domain of organizing conferences (the domain being well understandable for every researcher). Additionally, 'interesting correspondences' are also welcome. Results were evaluated automatically against reference alignments and by data-mining and logical reasoning techniques. Sample of correspondences and 'interesting correspondences' were also evaluated manually.

The directories and thesauri track proposed only web directories this years:

- **Directory (§6):** The directory real world case consists of matching web site directories (like open directory or Yahoo's). This year the track consists of two modalities, the first is composed by more than 4000 elementary tests, and the second is composed by a single test involving matching of two large directories (2854 and 6555 nodes each).

Instance matching (§7): The goal of the instance matching track is to evaluate the performance of different tools on the task of matching RDF individuals which originate from different sources but describe the same real-world entity. Instance matching is organized in two sub-tasks:

- **Data interlinking (DI)** Participants are requested to re-build the links among the available RDF resources. Reference alignments are provided for each resource as RDF alignments.
- **OWL data track (IIMB & PR):** In the OWL data track, data is provided as OWL individuals according to the RDF/XML format, while reference alignments are provided as RDF alignments. IIMB is divided into test cases and reference alignments are automatically generated by introducing controlled modifications in an initial reference ontology instance. Persons-Restaurants (PR) is a small real data test case where participants are requested to run matching tools against two collections of data concerning persons (person1 and person2) and one collection about restaurants (restaurant1).

The Benchmark, Anatomy and Conference datasets have been evaluated using the SEALS service. The reason for this is twofold: on the one hand, these data sets are well known to the organizers and have been used in many evaluations, contrary to the instance matching data sets, for instance. On the other hand, these data sets come with a high quality reference alignment which allows for computing the compliance based measures, such as precision and recall.

This year we had to cancel the VLCR (very large crosslingual resources) task since it had not enough participants to be retained. The Single task modality in the Directory track was also canceled due to lack of resources needed to cross check the reference alignments.

Table 1 summarizes the variation in the results expected from the tests under consideration.

test	formalism	relations	confidence	modalities	language
benchmarks	OWL	=	[0 1]	open	EN
anatomy	OWL	=	[0 1]	open	EN
conference	OWL-DL	=, <=	[0 1]	blind+open	EN
directory	OWL	=	1	blind+open	EN
di	RDF	=	[0 1]	open	EN
iimb	RDF	=	[0 1]	open	EN
vlcr	SKOS +OWL	exact-, closeMatch	[0 1]	blind expert	DU+EN

Table 1. Characteristics of the test cases (open evaluation is made with already published reference alignments and blind evaluation is made by organizers from reference alignments unknown to the participants).

2.2 The SEALS evaluation service

This year, participants have used the SEALS evaluation service for testing their systems and for launching their own evaluation experiments. A first version of this evaluation service[4] is based on the use of a web service interface wrapping the functionality of a matching tool to be evaluated [19]. Participants were invited to extend a web service interface[5] and deploy their matchers as web services, which are accessed during the evaluation process. This setting allows participants for debugging their systems, running their own evaluations and manipulating the results immediately in a direct feedback cycle.

In order to start an evaluation, participants had to specify the URL of the matcher service and the name of the matching system to be evaluated. Then they had to select the evaluation task to be used (Anatomy, Benchmark or Conference). The specified web service is validated by the system (two simple ontologies are used to check if the matcher generates alignments in the correct format). In case of problems, the concrete validation error is displayed to the user as direct feedback. In case of a successfully completed validation, the system returns a confirmation message and continues with the evaluation process. The values of precision, recall and F–measure are then displayed for each test case.

Furthermore, organizers have a tool for accessing the results registered for the campaign as well as all evaluations being carried out in the evaluation service. Specifically, results can be visualized and manipulated via an OLAP interface (Figure 1).

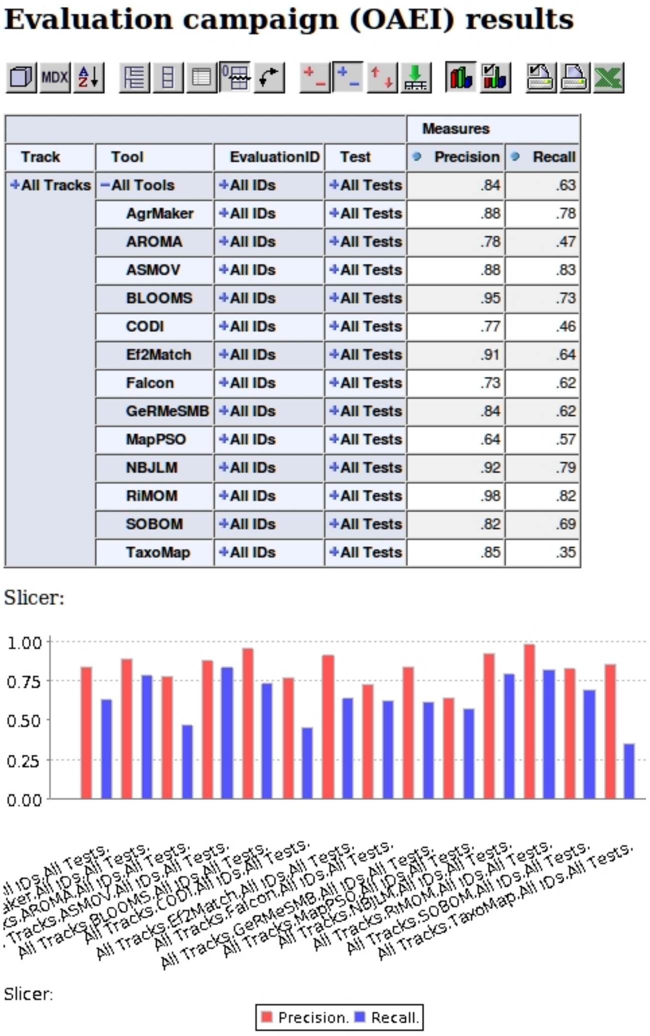
2.3 Preparatory phase

Ontologies to be matched and (where applicable) reference alignments have been provided in advance during the period between June 1^{st} and June 21^{st}, 2010. 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 8^{th}. The data sets did not evolve after this period.

---

[4] `http://seals.inrialpes.fr/platform/`

[5] `http://alignapi.gforge.inria.fr/tutorial/tutorial5/`



Evaluation campaign (OAEI) results

				Measures	
Track	Tool	EvaluationID	Test	Precision	Recall
+All Tracks	−All Tools	+All IDs	+All Tests	.84	.63
	AgrMaker	+All IDs	+All Tests	.88	.78
	AROMA	+All IDs	+All Tests	.78	.47
	ASMOV	+All IDs	+All Tests	.88	.83
	BLOOMS	+All IDs	+All Tests	.95	.73
	CODI	+All IDs	+All Tests	.77	.46
	Ef2Match	+All IDs	+All Tests	.91	.64
	Falcon	+All IDs	+All Tests	.73	.62
	GeRMeSMB	+All IDs	+All Tests	.84	.62
	MapPSO	+All IDs	+All Tests	.64	.57
	NBJLM	+All IDs	+All Tests	.92	.79
	RiMOM	+All IDs	+All Tests	.98	.82
	SOBOM	+All IDs	+All Tests	.82	.69
	TaxoMap	+All IDs	+All Tests	.85	.35

Slicer:

Fig. 1. Using OLAP for results visualization.

2.4 Preliminary tests

In this phase, participants were invited to test their systems in order to ensure that the systems can load the ontologies to be matched and generate the alignment in the correct format, namely the Alignment format expressed in RDF/XML [5]. Participants have been requested to provide (preliminary) results by August 30^{th}.

For the SEALS modality, testing could be conducted using the evaluation service while for the other tracks participants submitted their preliminary results to the organizers, who analyzed them semi-automatically, often detecting problems related to the format or to the naming of the required resulting files.

2.5 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.

For the standard OAEI modalities, participants had to run their systems on their own machines and submit the results via mail to the organizers. SEALS participants ran their systems via the SEALS evaluation service. They obtained a direct feedback on the results and could validate them as final results. Furthermore, SEALS participants were invited to register their tools by that time in the SEALS portal[6].

Participants also provided the papers that are published hereafter and a link to their systems and their configuration parameters.

2.6 Evaluation phase

In the evaluation phase, the organizers have evaluated the alignments provided by the participants and returned comparisons on these results. Final results were due by October 4^{th}, 2010. In the case of blind tests, only the organizers did the evaluation with regard to the withheld reference alignments.

Concerning SEALS, the participants have used the evaluation service for registering their results for the campaign. The evaluation effort is minimized due to the fact that the results are automatically computed by the services in the evaluation service and organizers have tools for manipulating and visualizing the results.

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 for compensating the lack of complete reference alignments.

2.7 Comments on the execution

Since a few years, the number of participating systems has remained roughly stable: 4 participants in 2004, 7 in 2005, 10 in 2006, 17 in 2007, 13 in 2008, 16 in 2009 and 15 in 2010.

---

[6] `http://www.seals-project.eu/join-the-community/`

The number of covered runs has decreased more than expected: 48 in 2007, 50 in 2008, 53 in 2009, and 37 in 2010. This may be due to the increasing specialization of tests: some systems are specifically designed for instance matching or for anatomy.

This year many of the systems are validated through web services thanks to the SEALS evaluation service. For the next OAEI campaign, we expect to be able to actually run the matchers in a controlled evaluation environment, in order to test their portability and deployability. This will also allow for comparing system on a same execution basis.

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

System	AgrMaker	AROMA	ASMOV	BLOOMS	CODI	Ef2Match	Falcon-AO	GeRMeSMB	LNR2	MapPSO	NBJLM	ObjectRef	RiMOM	SOBOM	TaxoMap	Total=15
Confidence	√	√	√	√		√	√	√	√	√	√	√	√	√	√	
benchmarks	√	√	√		√	√	√	√		√			√	√	√	11
anatomy	√		√	√	√	√		√			√			√	√	9
conference	√	√	√		√	√	√	√						√		8
directory			√					√		√					√	4
di												√	√			2
iimb+pr			√		√				√			√	√			5
Total	3	2	5	1	4	3	2	4	1	2	1	1	2	3	3	37

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.

Participants may be divided in two main categories: those who participated in the instance matching track and those who participated in ontology matching tracks. Three systems (ASMOV, CODI, RiMOM) participated in both types of tracks. Last year only two systems (DSSim and RiMOM) had 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 data set is to provide a stable and detailed picture of each algorithm. For that purpose, algorithms are run on systematically generated test cases.

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 one goal of these tests is to offer a permanent benchmark to be used by many, the test is an extension of the 2004 EON Ontology Alignment Contest, whose test numbering is (almost) fully preserved.

The tests are roughly the same as last year. 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

Eleven systems have participated in the benchmark track of this year's campaign (see Table 2). Four systems that had participated last year (AFlood, DSSim, Kosimap and Lily) did not participate this year, while two new systems (CODI and Ef2Match) have registered their results.

Table 3 shows the results, by groups of tests. For comparative purposes, the results of systems that have participated 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 full results are on the OAEI web site.

As shown in Table 3, two systems achieve top performances: ASMOV and RiMOM, with AgrMaker as a close follower, while SOBOM, GeRMeSMB and Ef2Match, re-

system	refalign		edna		AgrMaker		AROMA		ASMOV		CODI		Ef2Match		Falcon		GeRMeSMB		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.
1xx	1.00	1.00	1.00	1.00	0.98	1.00	1.00	0.98	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.34
2xx	1.00	1.00	0.43	0.57	0.95	0.84	0.94	0.46	0.99	0.89	0.83	0.42	0.98	0.63	0.81	0.63	0.96	0.66	0.67	0.59	0.99	0.83	0.97	0.74	0.86	0.29
3xx	1.00	1.00	0.51	0.65	0.88	0.58	0.83	0.58	0.88	0.84	0.95	0.45	0.92	0.75	0.89	0.76	0.90	0.42	0.72	0.39	0.94	0.76	0.79	0.75	0.71	0.32
H-mean	1.00	1.00	0.45	0.58	0.95	0.84	0.94	0.48	0.98	0.89	0.84	0.44	0.98	0.65	0.82	0.65	0.96	0.67	0.68	0.60	0.99	0.84	0.97	0.75	0.86	0.29
Relaxed	1.00	1.00	0.45	0.58	0.95	0.84	0.94	0.48	0.99	0.89	0.84	0.44	0.98	0.65	0.82	0.65	0.96	0.67	0.68	0.60	0.99	0.84	0.97	0.75	0.86	0.29
Weighted	1.00	1.00	0.68	0.57	0.95	0.83	0.94	0.42	0.98	0.61	0.84	0.44	0.98	0.64	0.96	0.46	0.96	0.64	0.86	0.56	0.99	0.83	0.98	0.37	0.87	0.28

Table 3. Results obtained by participants on the benchmark test case (harmonic means). Relaxed precision and recall correspond to the three measures of [4]: symmetric proximity, correction effort and oriented. The same results have been obtained using these three measures. Weighted precision and recall takes into account the confidence associated to correspondence by the matchers.

spectively, had presented intermediary values of precision and recall. In the 2009 campaign, Lily and ASMOV had the best results, with aflood and RiMOM as followers, while GeRoME, AROMA, DSSim and AgrMaker had intermediary performance. The same group of matchers has been presented in both campaigns. No system had strictly lower performance than edna.

Looking for each group of tests, in simple tests (1xx) all systems have similar performance, excluding TaxoMap which has presented low value of recall. As noted in previous campaigns, the algorithms have their best score with the 1xx test series. This is because there are no modifications in the labels of classes and properties in these tests and basically all matchers are able to deal with the heterogeneity in labels. For systematic tests (2xx), which allow better to distinguish the strengths of algorithms, ASMOV and RiMOM, respectively, achieve the best results, followed by AgrMaker, SOBOM, GeRMeSMB and Ef2Match, respectively, which have presented good performance, specially in terms of precision. Finally, for real cases (3xx), ASMOV (in average) provided the best results, with RiMOM and Ef2Match as followers. The best precision for these cases was obtained by the new participant CODI.

In general, systems have improved their performance since last year: ASMOV and RiMOM improved their overall performance, AgrMaker and SOBOM significantly improved their recall, while MapPSO and GeRMeSBM improved precision. Only AROMA has significantly decreased in recall. There is no unique set of systems achieving the best results for all cases, which indicates that systems exploiting different features of ontologies perform accordingly to the features of each test case.

The results have also been compared with the relaxed measures proposed in [4], namely symmetric proximity, correction effort and oriented measures ("Relaxed measures" in Table 3). They are different generalisations of precision and recall in order to better discriminate systems that slightly miss the target from those which are grossly wrong. We have used strict versions of these measures (as published in [4] and contrary to previous years). As Table 3 shows, there is no improvement when comparing classical and relaxed precision and recall. This can be explained by the fact that participating algorithms miss the target, by relatively far (the false negative correspondences found by the matchers are not close to the correspondences in the reference alignment) so the gain provided by the relaxed measures has no impact.

We have introduced experimentally confidence-weighted precision and recall in which correspondences are weighted by the confidence matchers put on it. If the confidence is 1., then the correspondence scores exactly like in classical precision and recall. Otherwise, it scores for the amount of confidence. If the correspondence is correct, this will contribute to decrease recall – it will be counted for less than 1. –, if the correspondence is incorrect, this will increase precision – by counting the mistake for less than 1. So this rewards systems able to provide accurate confidence measures (or penalizes less mistakes on correspondences with low confidence). These measures provide precision increase for Falcon, MaPSO and edit distance (which had apparently many incorrect correspondences with low confidence), and recall decrease for Falcon, ASMOV and SOBOM (which had apparently many correct correspondences with low confidence). There are only little variation for other systems. As expected, CODI, which confidence was always 1, shows no variation.

As last year, many algorithms provided their results with confidence measures. It is thus possible to draw precision/recall graphs in order to compare them. 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). Contrary to previous years these graphs are not drawn with the same principles as TREC's. They now show the real precision at n% recall and they stop when no more correspondences are available (then the end point corresponds to the precision and recall reported in Table 3). The values are not anymore an average but a real precision and recall over all the tests. The numbers in the legend are the Mean Average Precision (MAP): the average precision for each correct retrieved correspondence. These new graphs represent well the effort made by the participants to keep a high precision in their results, and to authorize a loss of precision with a few correspondences with low confidence.

The results presented in Table 3 and those displayed in Figure 2 single out the same group of systems, ASMOV, RiMOM and AgrMaker, which perform these tests at the highest level. Out of these, ASMOV has slightly better results than the two others. So, this confirms the previous observations on raw results.

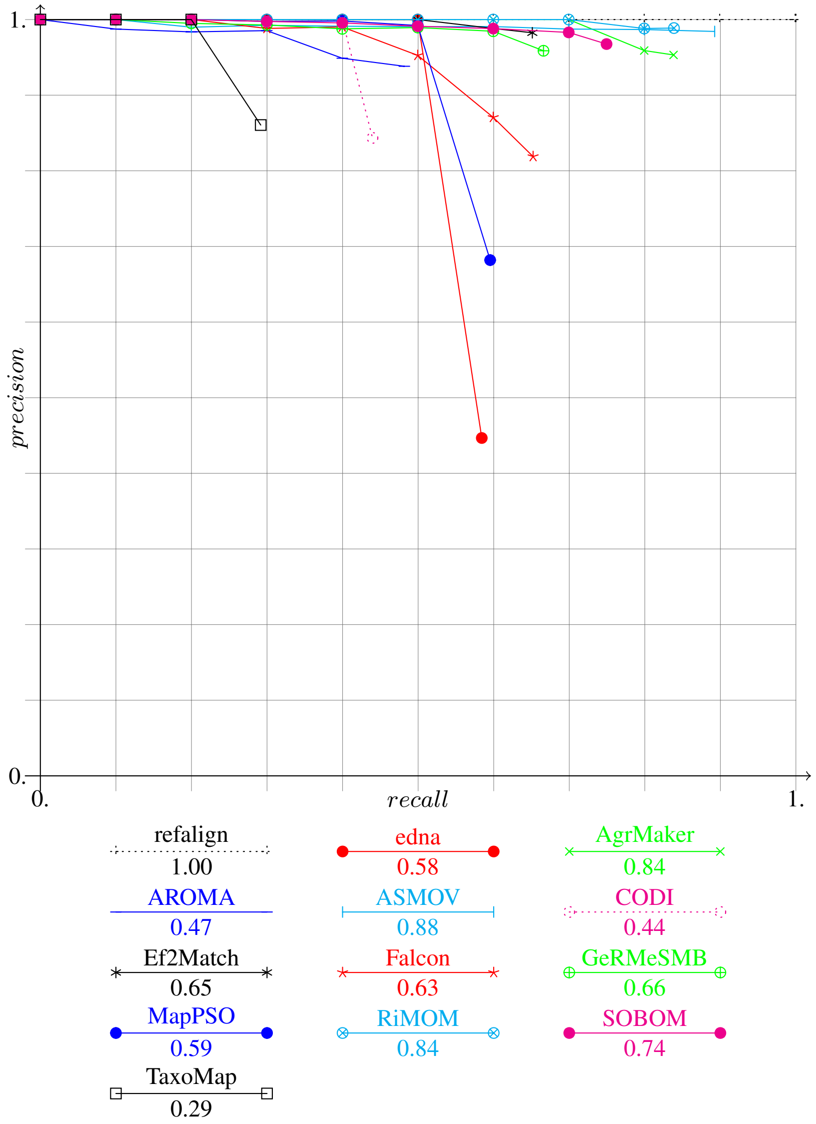


Fig. 2. Precision/recall graphs for benchmarks. The results given by the participants are cut under a threshold necessary for achieving $n\%$ recall and the corresponding precision is computed. Systems for which these graphs are not meaningful (because they did not provide graded confidence values) are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

4 Anatomy

The anatomy track confronts matching technology with a specific type of ontologies from the biomedical domain. In this domain, a significant number of ontologies have been built covering different aspects of medical research.

4.1 Test data and experimental setting

The data set of this track has been used since 2007. For a detailed description we refer the reader to the OAEI 2007 [7] results paper. As in previous years, we divided the matching task into four subtasks. Subtask #1 is compulsory for participants of the anatomy track, while subtask #2, #3 and #4 are again optional tasks.

Subtask #1 The matcher has to be applied with its standard settings.
Subtask #2 An alignment has to be generated that favors precision over recall.
Subtask #3 An alignment has to be generated that favors recall over precision.
Subtask #4 A partial reference alignment has to be used as additional input.

Notice that in 2010 we used the SEALS evaluation service for subtask #1. In the course of using the SEALS services, we published the complete reference alignment for the first time. In future, we plan to include all subtasks in the SEALS modality. This requires to extend the interfaces of the SEALS evaluation service to allow for example an (incomplete) alignment as additional input parameter.

The harmonization of the ontologies applied in the process of generating a reference alignment (see [2] and [7]) resulted in a high number of rather trivial correspondences (61%). These correspondences can be found by very simple string comparison techniques. At the same time, we have a good share of non-trivial correspondences (39%). This is an important characteristic of the data set to be taken into account in the following analysis. The partial reference alignment used in subtask #4 is the union of all trivial correspondences and 54 non-trivial correspondences.

We slightly improved the test data set for the 2010 evaluation. We removed some doubtful subsumption axioms and added a number of disjointness statements at the top of the hierarchies to increase the expressivity of the data set. Furthermore, we eliminated some incorrect correspondences.[7]

In previous years, we reported about runtimes measured by the participants. The differences we observed – from several minutes to several days – could not be explained by the use of different hardware. However, these differences became less significant over the years. Therefore, we abstained from an analysis of runtimes this year. In 2011, we plan to execute the matching systems on the SEALS platform to enable an exact measurement of runtimes not biased by differences in hardware equipment. So far we refer the reader interested in runtimes to the result papers of the participants.

---

[7] We gratefully thank Elena Beisswanger (Jena University Language and Information Engineering Lab) for her thorough support on improving the quality of the data set. The modifications are documented at `http://webrum.uni-mannheim.de/math/lski/anatomy10/modifications2010.html`

4.2 Results

While the number of participants has been roughly stable over four years, we had in 2010 more systems that participated for the first time (5 systems) than in previous years (in average 2 systems; see Table 4 for an overview). Three of the newcomers participate also in other tracks, while NBJLM and BLOOMS participate only in the Anatomy track. Notice also that AgreementMaker (AgrMaker) uses a track-specific parameter setting. Taking part in several tracks with a standard setting makes it obviously much harder to obtain good results in a specific track.

System	**2007**	**2008**	**2009**	**2010**
AFlood		√	√	
AgrMaker	√		+	+
AROMA		√	√	
AOAS	+			
ASMOV	√	√	√	√
BLOOMS				+
CODI				√
DSSim	√	√	√	
Ef2Match				+
Falcon AO	√			
GeRMeSMB				√
Kosimap			√	
Lily	√	√	√	
NBJLM				+
Prior+	√			
RiMOM	√	+	√	
SAMBO	+	+		
SOBOM			+	+
TaxoMap	√	√	√	+
X SOM	√			
Avg. F-measure	0.598	0.718	0.764	0.785

Table 4. Overview on anatomy participants from 2007 to 2010, a $\surd$-symbol indicates that the system participated, + indicates that the system achieved an F-measure ≥ 0.8 in subtask #1.

In the last row of Table 4, the average of F-measures per year in subtask #1 is shown. We observe significant improvements over time. However, the measured improvements decrease over time and seem to reach a top (2007 +12% $\rightarrow$ 2008 +5% $\rightarrow$ 2009 +2% $\rightarrow$ 2010). We have marked the participants with an F-measure ≥ 0.8 with a + symbol. Note that in each of the previous years, only two systems reached this level, while in 2010 six systems reached a higher value than 0.8.

Main results for subtask #1. The results for subtask $\#1$ are presented in Table 5 ordered with respect to the achieved F-measure. In 2010, AgreementMaker (AgrMaker) generated the best alignment with respect to F-measure. This result is based on a high recall compared to the systems on the following positions. Even the SAMBO system of 2007 could not generate a higher recall with the use of UMLS.

	Task #1			Task #2			Task #3			Recall+	
System	**Prec.**	**F**	**Rec.**	**Prec.**	**F**	**Rec.**	**Prec.**	**F**	**Rec.**	**#1**	**#3**
AgrMaker*	0.903	0.877	0.853	0.962	0.843	0.751	0.771	0.819	0.874	0.630	0.700
Ef2Match	0.955	0.859	0.781	0.968	0.842	0.745	0.954	0.859	0.781	0.440	0.440
NBJLM*	0.920	0.858	0.803	-	-	-	-	-	-	0.569	-
SOBOM	0.949	0.855	0.778	-	-	-	-	-	-	0.433	-
BLOOMS	0.954	0.828	0.731	0.967	0.829	0.725	-	-	-	0.315	-
TaxoMap	0.924	0.824	0.743	0.956	0.801	0.689	0.833	0.802	0.774	0.336	0.414
ASMOV	0.799	0.785	0.772	0.865	0.808	0.757	0.717	0.753	0.792	0.470	0.538
CODI	0.968	0.779	0.651	0.964	0.785	0.662	0.782	0.736	0.695	0.182	0.383
GeRMeSMB	0.884	0.456	0.307	0.883	0.456	0.307	0.080	0.147	0.891	0.249	0.838

Table 5. Results for subtasks #1, #2 and #3 in terms of precision, F-measure, and recall (in addition recall+ for #1 and #3). Systems marked with a * do not participate in other tracks or have chosen a setting specific to this track. Note that ASMOV modified its standard setting in a very restricted way (activating UMLS as additional resource). Thus, we did not mark this system.

AgreementMaker is followed by three participants (Ef2Match, NBJLM and SOBOM) that clearly favor precision over recall. Notice that these systems obtained better scores or scores that are similar to the results of the top systems in the previous years. One explanation can be seen in the fact that the organizers of the track made the reference alignment available to the participants. More precisely, participants could at any time compute precision and recall scores via the SEALS services to test different settings of their algorithms. This allows to improve a matching system in a direct feedback cycle, however, it might happen that a perfect configuration results in problems for different data sets.

Recall+ and further results. We use again the recall+ measure as defined in [7]. It measures how many non trivial correct correspondences, not detectable by string equivalence, can be found in an alignment. The top three systems with respect to recall+ regarding subtask #1 are AgreementMaker, NBJLM and ASMOV. Only ASMOV has participated in several tracks with the same setting. Obviously, it is not easy to find a large amount of non-trivial correspondences with a standard setting.

In 2010, six systems participated in subtask #3. The top three systems regarding recall+ in this task are GeRoMe-SMB (GeRMeSMB), AgreementMaker and ASMOV. Since a specific instruction about the balance between precision and recall is missing in the description of the task, the results vary to a large degree. GeRoMe-SMB detected 83.8% of the correspondences marked as non trivial, but at a precision of 8%. AgreementMaker and ASMOV modified their settings only slightly, however, they were still able to detect 70% and 53.8% of all non trivial correspondences.

In subtask #2, seven systems participated. It is interesting to see that systems like ASMOV, BLOOMS and CODI generate alignments with slightly higher F-measure for this task compared to the submission for subtask #1. The results for subtask #2 for AgreementMaker are similar to the results submitted by other participants for subtask #1. This shows that many systems in 2010 focused on a similar strategy that exploits the specifics of the data set resulting in a high F-measure based on a high precision.

Subtask #4. In the following, we refer to an alignment generated for subtask #n as A_n. In our evaluation we use again the method introduced in 2009. We compare both $A_1 \cup R_p$ and $A_4 \cup R_p$ with the reference alignment R.[8] Thus, we compare the situation where the partial reference alignment is added after the matching process against the situation where the partial reference alignment is available as additional resource exploited within the matching process. Note that a direct comparison of A_1 and A_4 would not take into account in how far the partial reference alignment was already included in A_1 resulting in a distorted interpretation.

System	**Δ-Precision**	**Δ-F-measure**	**Δ-Recall**
AgrMaker	$+0.025_{\ 0.904\rightarrow 0.929}$	$-0.002_{\ 0.890\rightarrow 0.888}$	$-0.025_{\ 0.876\rightarrow 0.851}$
ASMOV	$+0.029_{\ 0.808\rightarrow 0.837}$	$+0.006_{\ 0.816\rightarrow 0.822}$	$-0.016_{\ 0.824\rightarrow 0.808}$
CODI	$-0.002_{\ 0.970\rightarrow 0.968}$	$+0.019_{\ 0.824\rightarrow 0.843}$	$+0.030_{\ 0.716\rightarrow 0.746}$
SAMBOdtf$_{2008}$	$+0.021_{\ 0.837\rightarrow 0.856}$	$+0.011_{\ 0.852\rightarrow 0.863}$	$+0.003_{\ 0.867\rightarrow 0.870}$

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

Results are presented in Table 6. Three systems participated in task #4 in 2010. Additionally, we added a row for the 2008 submission of SAMBOdtf. This system had the best results measured in the last years. AgreementMaker and ASMOV use the input alignment to increase the precision of the final result. At the same time these systems filter out some correct correspondences, finally resulting in a slightly increased F-measure. This fits with the trend observed in the past years (compare with the results for SAMBOdtf in 2008). The effects of this strategy are not very strong. However, as argued in previous years, the input alignment has a characteristics that makes hard to exploit this information. CODI has chosen a different strategy. While changes in precision are negligible, recall increases by 3%. Even though the overall effect is still not very strong, the system exploits the input alignment in the most effective way. However, the recall of CODI for subtask #1 is relatively low compared to the other systems. It is unclear whether the strategy of CODI would also work for the other systems where a ceiling effect might prevent the exploitation of the positive effects.

4.3 Conclusions

Overall, we see a clear improvement comparing 2010 results with the results of previous years. This holds both for the "average participant" as well as for the top performer. A very positive outcome can be seen in the increased recall values. In addition to the evaluation experiments reported so far, we computed the union of all submissions to subtask #1. For the resulting alignment we measured a precision of 69.7% and a recall of 92.7%. We added additionally the correct correspondences generated in subtask #3 and reached a a recall of 97.1%. Combining the strategies used by different matching systems it is thus possible to detect nearly all correct correspondences.

---

[8] We use $A_4 \cup R_p$ – instead of using A_4 directly – to ensure that a system, which does not include the input alignment in the output, is not penalized.

5 Conference

The conference test set introduces matching several moderately expressive ontologies. Within this track, participant results were evaluated using diverse evaluation methods. The evaluation has been supported by the SEALS evaluation service this year.

5.1 Test data

The data set of this track has been extended with one ontology being in OWL 2.[9] For a data set description we refer the reader to the OAEI 2009 results paper [6].

5.2 Results

We had eight participants: AgreementMaker (AgrMaker), AROMA, ASMOV, CODI, Ef2Match, Falcon, GeRMeSMB and SOBOM. All participants delivered all 120 alignments. CODI delivered ‘certain’ correspondences, the other matchers delivered correspondences with graded confidence values between 0 and 1.

Evaluation based on the reference alignments. We evaluated the results of participants against reference alignments. They include all pairwise combinations between 7 different ontologies, i.e. 21 alignments.

matcher	confidence threshold	Prec.	FMeas.	Rec.
AgrMaker	0.66	.53	.58	.62
AROMA	0.49	.36	.42	.49
ASMOV	0.22	.57	.60	.63
CODI	*	.86	.62	.48
Ef2Match	0.84	.61	.60	.58
Falcon	0.87	.74	.59	.49
GeRMeSMB	0.87	.37	.43	.51
SOBOM	0.35	.56	.56	.56

Table 7. Confidence threshold, precision and recall for optimal F-measure for each matcher.

For a better comparison, we established the confidence threshold which provides the highest average F-measure (Table 7). Precision, recall, and F-measure are given for this optimal confidence threshold. The dependency of F-measure on the confidence threshold can be seen from Figure 3. There is one asterisk in the column of confidence threshold for matcher CODI which did not provide graded confidence.

In conclusion, the matcher with the highest average F-measure (62%) is CODI which did not provide graded confidence values. Other matchers are very close to this score (e.g. ASMOV with F-Measure 0.60, Ef2Match with F-Measure 0.60, Falcon with F-Measure 0.59). However, we should take into account that this evaluation has been made over a subset of all alignments (one fifth).

---

[9] Ontologies have been developed within the OntoFarm project `http://nb.vse.cz/~svatek/ontofarm.html`

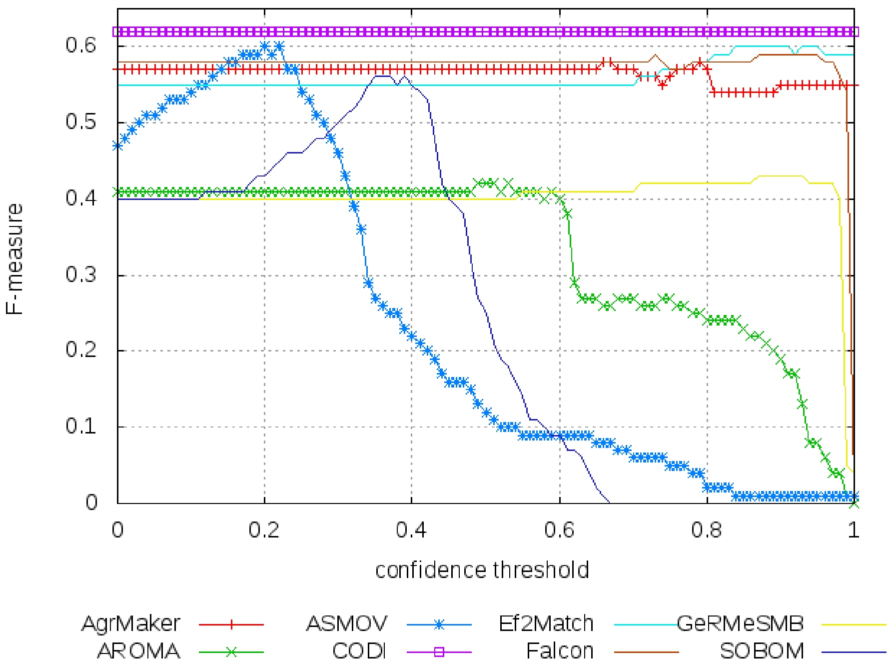


Fig. 3. F-measures depending on confidence.

Comparison with previous years. Three matchers also participated in the last two years. ASMOV participated in all three consecutive years with increasing highest average F-measure: from .43 in 2008 and .47 in 2009 to .60 in 2010. AgreementMaker participated with .57 in 2009 and with .58 in 2010 regarding highest average F-measure. Finally, AROMA participated with the same highest average F-measure in both 2009 and 2010.

Evaluation based on posterior manual labeling. This year we took the most secure correct correspondences, i.e., with the highest confidence, as a population for each matcher. Per matcher, we evaluated 100 correspondences randomly chosen from all correspondences of all 120 alignments with confidence 1.0 (sampling). Because AROMA, ASMOV, Falcon, GeRMeSMB and SOBOM do not have enough correspondences with 1.0 confidence we took the 100 correspondences with highest confidence. For all of these matchers (except ASMOV where we found exactly 100 correspondences with highest confidence values) we sampled over their population.

Table 8 presents approximated precisions for each matcher over its population of best correspondences. N is the population of all best correspondences for one matcher. n is a number, ideally 100, of randomly chosen correspondences, among the 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 [20].

matcher	AgrMaker	AROMA	ASMOV	CODI	Ef2Match	Falcon	GeRMeSMB	SOBOM
N	804	108	100	783	1236	127	110	105
n	100	100	100	100	100	100	100	100
TP	92	68	86	98	79	96	30	82
P*	.92 ±9.4	.68 ±2.7	.86	.98 ±9.3	.79 ±9.6	.96 ±4.6	.30 ±3.0	.82 ±2.2

Table 8. Approximated precision for 100 best correspondences for each matcher.

From Table 8 we can conclude that CODI, Falcon and AgreementMaker have the best precision (higher than 90%) over their 100 more confident correspondences.

Evaluation based on data mining supported with matching patterns. Results of OAEI participants, i.e. correspondences, contain several attributes characterizing these correspondences from different aspects. Additionally, there is also information in which matching patterns the given correspondence participates; for details about this see [15; 22].

In total there are nine matching patterns ($MP1$ - $MP9$). $MP4$, $MP5$, and $MP6$ are inspired by correspondence patterns from [16]. In principle, it is not possible to say which matching 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 (see section below).

	Antecedent				Succedent	Values	
	System	Certainty factor	Resource1	Resource2	Result	Supp	AvgDff
t1	AgrMaker	$< 1.0; 1.0 >$	*	*	+	0.024	0.95
t2	ASMOV	$< 0.4; 0.8)$	*	*	+	0.01	0.6
t3	SOBOM	< 0.4	*	*	+	0.024	0.37
t4	ASMOV	< 0.4	*	w	-	0.013	1.25
t5	SOBOM	*	*	w	-	0.014	1.21

Table 9. Hypotheses for tasks 1 and 2.

	Antecedent		Succedent	Values
	System	ResultMP	Supp	AvgDff
m1	ASMOV	MP2	0.013	0.79
m2	AgrMaker	MP2	0.015	0.1
m3	AROMA	MP4	0.016	0.16

Table 10. Association hypotheses related to matching patterns.

For data mining we employed the *4ft-Miner* procedure of the *LISp-Miner* data mining system[10] for mining of *association rules*. We found several interesting association

---

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

hypotheses: $t1$ to $t5$ are related to confidence value or underlying resources of ontologies (see Table 9) and $m1$ to $m3$ are related to matching patterns (see Table 10). In total, there were 16522 correspondences in the data matrix. For instance we can interpret *hypothesis t1* as follows *correspondences that are produced by AgrMaker and have the highest confidence value (i.e. 1.0) are by 95% (i.e. almost twice) more often correct than correspondences produced by all systems with all confidence values (on average).*

In conclusion, regarding first three hypotheses we could say that AgrMaker is quite sure about correspondences with the highest value than other matchers. On the other side, ASMOV is surprisingly correct about correspondences with lower confidence values than other matchers. These hypotheses confirmed findings from the previous year since these two matchers also participated in the OAEI-2009. SOBOM is more correct for correspondences with the lowest confidence values. Hypotheses t4 and t5 point out that ASMOV and SOBOM work worse with ontologies based on the web. Regarding hypotheses containing matching patterns, ASMOV and AgrMaker found MP2, while AROMA found MP4 more often than other systems. In comparison with the previous year, there are no interesting hypotheses with matching patterns related to inconsistency. This can be explained by the fact that many occurrences of matching patterns related with inconsistency is generally lower than in the previous year of OAEI. These findings roughly correspond with the results of the evaluation based on the alignment coherence in the next section.

Evaluation based on alignment coherence. In the following we apply the Maximum Cardinality measure as proposed in [14] to measure the degree of alignment incoherence. Results are depicted in Table 11 which shows the average for all testcases of the conference track except the testcases where the ontologies confious and linklings are involved. These ontologies resulted in some combinations of ontologies and alignments in reasoning problems. Note that we did not use the original alignments, but the alignments with optimal threshold. However, the average size of the resulting alignment still varies to a large degree.

Matcher	AgrMaker	AROMA	ASMOV	CODI	Ef2Match	Falcon	GeRMeSMB	SOBOM
Max-Card %	>14.8%	>17.5%	5.6%	0.1%	7.2%	>4.8%	>12.6%	>10.7
N	17.1	16.4	18.2	6.9	12.8	8.9	18.2	11.7

Table 11. Degree of incoherence and size of alignment in average for the optimal a posteriori threshold. The prefix > is added whenever the search algorithm stopped in one of the testcase due to a timeout of 1000 seconds prior to finding the solution.

Compared to the other participants CODI generates the lowest degree of incoherence. This result is partially caused by the small size of alignments that make the occurrence of an incoherence less probable. Taking this into account, the ASMOV system achieves a remarkable result. Even though the alignments of ASMOV comprise the highest number of correspondences, the degree of incoherence 5.6% is relatively small due to the verification component built into the system [12]. Overall it is a suprising result that still only few matching systems take alignment incoherence into account.

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". This task focusses on evaluating the performances of existing matching tools in a real world taxonomy integration scenario.

6.1 Test set

As in previous years, the data set exploited in the directory matching task was constructed from the Google, Yahoo and Looksmart web directories following the methodology described in [10]. 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 `rdfs:subClassOf`.

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 [10], human annotators consider only 2265 correspondences instead of the full matching problem.

The specific characteristics of the data set are:

- Simple relationships. Basically web directories contain only one type of relationship called "classification relation".
- Vague terminology and modeling principles: The matching tasks incorporate the typical "real world" modeling and terminological errors.
- More than 4.500 node matching tasks, where each node matching task is composed of the paths to root of nodes in the web directories.
- Reference correspondences for the equivalence relation for all the matching tasks.

6.2 Results

In OAEI-2010, 3 out of 15 matching systems participated in the web directories test case, while in OAEI-2009 7 out of 16, in OAEI-2008, 7 out of 13, in OAEI-2007, 9 out of 17, in OAEI-2006, 7 out of 10, and in OAEI-2005, 7 out of 7 did it. The systems that submitted their results to the Single task modality of the Directory track were ASMOV, GeRoMe-SMB, MapPSO and TaxoMap, though the task was canceled due to lack of resources needed to cross check the reference alignments.

Precision, F-measure and recall results of the systems are shown in Figure 4. These indicators have been computed following the TaxMe2 [10] methodology, with the help of the Alignment API [5], version 4.0.

We can observe that ASMOV has maintained its recall, but increased its precision by 1 point in comparison to 2009. MapPSO has increased its recall (+27) and precision (+7) values, resulting in a 20 points increase in the F-measure from its last participation in 2008. TaxoMap has decreased its recall (-7) but increased its precision (+3), resulting

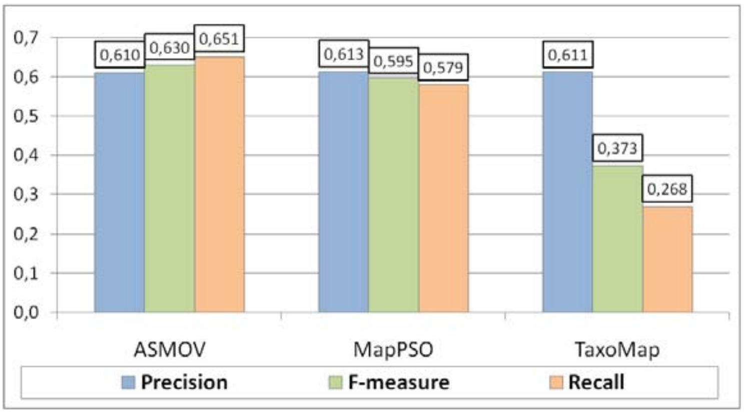


Fig. 4. Matching quality results.

in an overall decrease of F-measure (-6) from its last participation in 2009. ASMOV is the system with the highest F-measure value in 2010.

In total, 24 matching systems have participated during the 6 years (2005 - 2010) of the OAEI campaign in the directory track where 40 individual submissions from different systems have been received over the past 6 years. No single system has participated in all campaigns involving the web directory dataset (2005 - 2010). A total of 15 systems have participated only one time in the evaluation, 5 systems have participated 3 times (DSSIM, Falcon, Lily, RiMOM and TaxoMap), and only 1 system has participated 4 times (ASMOV).

As can be seen in Figure 5, this year there is an small increase (2%) in the average precision, in comparison to 2007 and 2008. The average recall in 2010 increased in comparison to 2009, reaching the same highest average recall value as in 2007. Considering F-measure, results for 2010 show the highest average in the 5 years (2006 to 2010). Notice that in 2005 the data set allowed only the estimation of recall, therefore Figure 5 does not contain values of precision and F-measure for 2005.

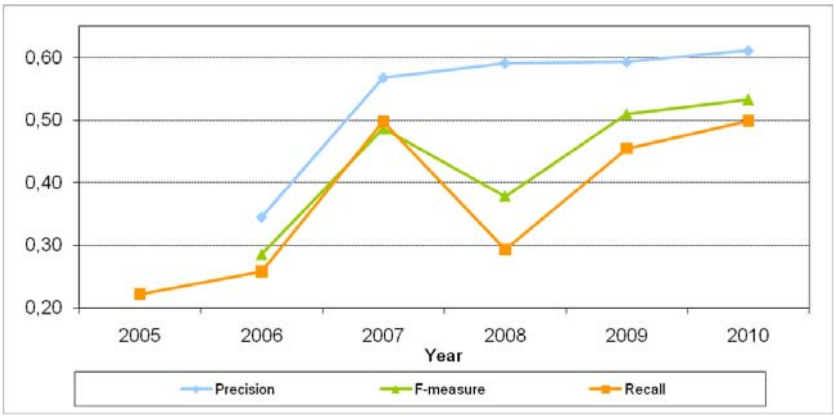


Fig. 5. Average results of the participating systems per year.

A comparison of the results from 2006 - 2010 for the top-3 systems of each year based on the highest values of the F-measure indicator is shown in Figure 6. An important note is that since there are only 3 participants this year, they all made their ways into the top three. The comparison of the top three participants has being made since 2006, therefore we keep the same comparison (and not the top 2, for example) for historical reasons. The quality of the best F-measure result of 2010 (0.63) achieved by ASMOV is equal to the best F-measure of 2009 by the same system, higher than the best F-measure of 2007 by DSSim (0.49) and than that of 2006 by Falcon (0.43), but still lower than the best F-measure of 2007 (0.71) by OLA_2. All three participating systems have achieved the same precision in 2010 (0.61), but this precision is lower than the best values of 2009 (0.62) by kosimap, in 2008 (0.64) by ASMOV and in 2007 by both OLA_2 and X-SOM. Finally, for what concerns recall, the best result of 2010 achieved by ASMOV (0.65) is equal to the best value of 2009 (0.65) also achieved by ASMOV, 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) by OLA_2.

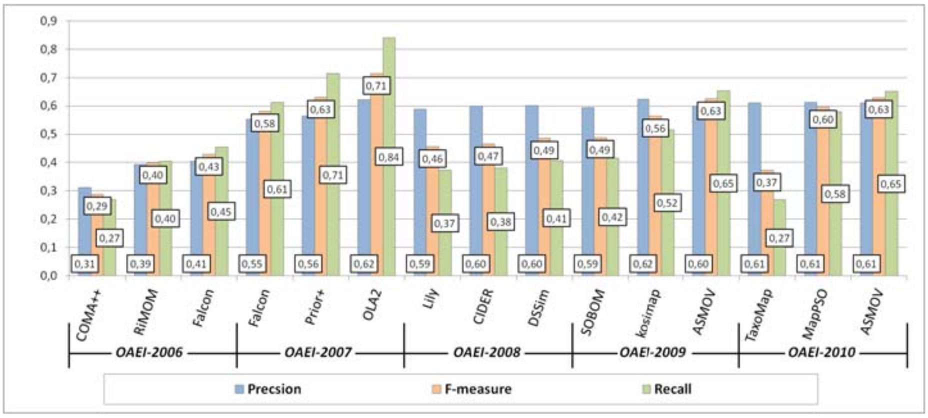
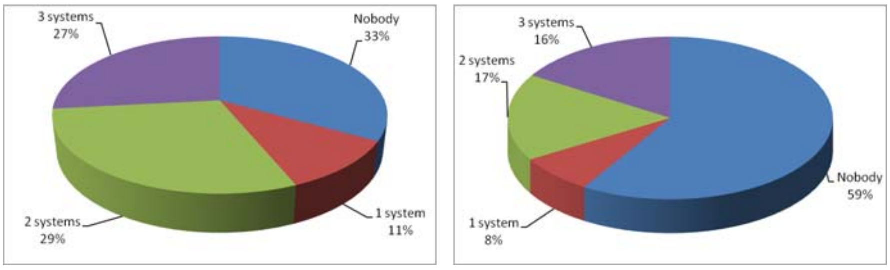


Fig. 6. Comparison of matching quality results in 2006 - 2010.

Partitions of positive and negative correspondences, according to the system results, are presented in Figure 7 **a)** and Figure 7 **b)**, respectively. Figure 7 **a)** shows that the systems managed to discover only 67% of the total number of positive correspondences (Nobody = 33%). Only 27% of positive correspondences were found by all three participating systems. The percentage of positive correspondences found by the systems this year is slightly lower than the values of 2009, when 68% of the positive correspondences where found [6], but still higher than the values of 2008, when 54% of the positive correspondences where found [3].

Figure 7 **b)** shows that more than half (59%) of the negatives correspondences were not found by the systems (correctly) in comparison to 56% not found in 2009). Figure 7 **b)** also shows that all participating systems found 16% of the negative correspondences, i.e., mistakenly returned them as positive, in comparison to 17% in 2009. These two observations explain the small increase in precision in Figure 5. The last two observations also suggest that the discrimination ability of the dataset remains as high as in previous years.



a) **b)**

Fig. 7. Partition of the system results: **a)** on positive correspondences, **b)** on negative correspondences.

Figure 7 **a)** shows that 33% 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%) but worse than of 2009 (32%). In 2007, all the positive correspondences have been collectively found; these results (2007) were exceptional because the participating systems altogether had a full coverage of the expected results and very high precision and recall. Unfortunately, the best systems of 2007 did not participate in the last years and the other systems do not seem to cope with the results of 2007.

Figure 7 **b)** shows that this year 59% of the negatives correspondences were correctly not found. There is an increase in comparison to the value of 2009 (56%) but 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 2010). This year 16% of the negative correspondences were mistakenly found by all the (3) participating systems, being the best value that of 2008 (1% for all (7) participating systems). An interpretation of these observations could be that the set of participating systems in 2010 seems to have found a good balance between being "cautious" (not finding negatives) and being "brave" (finding positives), resulting in average increases on precision, recall and F-measure as shown in Figure 5. In average, in 2010 the participants have a more "cautious" strategy of all years except 2008, being a little bit more "brave" than in 2007 and 2008. In 2007, we can observe that the set of 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 also 98% of the negative correspondences.

6.3 Comments

This year the average performance of the participants (given by the increase in precision and F-measure in Figure 5) is the best of all 5 years (2006 to 2010). This suggests that the set of participating systems has found a balance between a "brave and cautious" behavior for discovering correspondences. However, the value for the F-measure (0.53) indicates that there is still room for improvements. In comparison to 2009, there is an increase of 2% in F-measure where the average F-measure was (0.51). Finally, as partitions of positive and negative correspondences indicate (see Figure 7 **a)** and Figure 7 **b)**), the dataset still retains a good discrimination ability, i.e., different sets of correspondences are still hard for the different systems.

7 Instance matching

The instance matching track was included into the OAEI campaigns for the second time. The goal of the track is to evaluate the performance of different tools on the task of matching RDF individuals which originate from different sources but describe the same real-world entity. With the development of the Linked Data initiative, the growing amount of semantic data published on the Web and the need to discover identity links between instances from different repositories, this problem gained more importance in the recent years. Unlike the other tracks, the instance matching tests specifically focus on an ontology ABox. However, the problems which have to be resolved in order to match instances correctly can originate at the schema level (use of different properties and classification schemas) as well as at the data level, e.g., different format of values. This year, the track included two tasks. The first task (data interlinking - DI) aims at testing the performance of tools on large-scale real-world datasets published according to the Linked Data principles. The second one (IIMB & PR) uses a set of artificially generated and real test cases respectively. These are designed to illustrate all common cases of discrepancies between individual descriptions (different value formats, modified properties, different classification schemas). The list of participants to the Instance Matching track is shown in Table 12.

System	**DI**	**IIMB_SMALL**	**IIMB_LARGE**	**PR**
ASMOV		√	√	√
ASMOV_D				√
CODI		√	√	√
LN2R				√
ObjectCoref	√			√
RiMOM	√	√	√	√

Table 12. Participants in the instance matching track.

7.1 Data interlinking task (DI)

Data interlinking is known under many names according to various research communities: equivalence mining, record linkage, object consolidation and coreference resolution to mention the most used ones. In each case, these terms are used for the task of finding equivalent entities in or across datasets. As the quantity of datasets published on the Web of data dramatically increases, the need for tools helping to interlink resources becomes more critical. It is particularly important to maximize the automation of the interlinking process in order to be able to follow this expansion.

This year, we propose to interlink four datasets together. We have selected datasets for their potential to be interlinked, for the availability of curated interlinks between them, and for their size. All datasets are on the health-care domain and all of them contain information about drugs (see [13] for more details on the datasets):

dailymed is published by the US National Library of Medecine and contains information about marketed drugs. *Dailymed* contains information on the chemical structure, mechanism of action, indication, usage, contraindications and adverse reactions for the drugs.

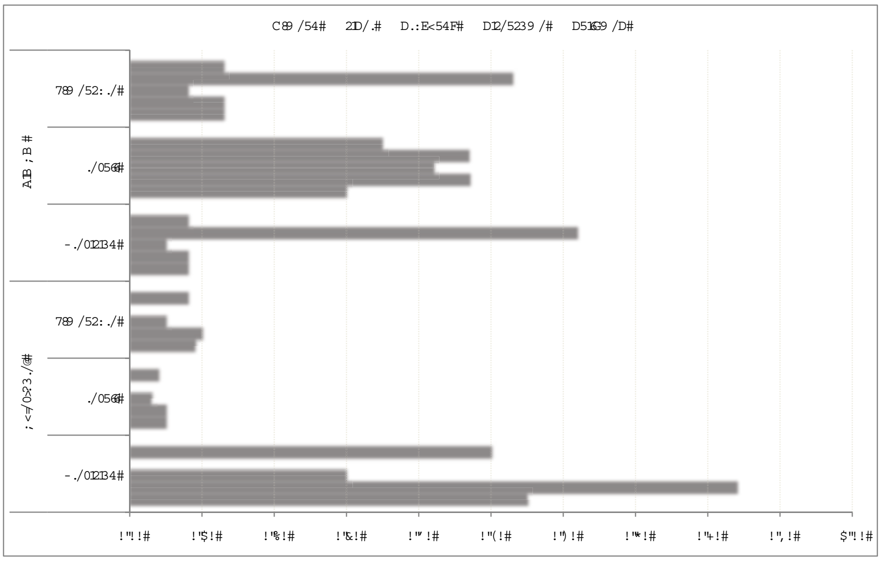


Fig. 8. Results of the DI task.

diseasome contains information about 4300 disorders and genes.

drugbank is a repository of more than 5000 drugs approved by the US Federal Drugs Agency. It contains information about chemical, pharmaceutical and pharmacological data along with the drugs data.

sider contains information on marketed drugs and their recorded adverse reactions. It was originally published on flat files before being converted as linked-data through a relational database.

These datasets were semi-automatically interlinked using Silk [21] and ODD Linker [11] providing the reference alignments for this task and participants were asked to retrieve these links using an automatic method.

Only two systems participated in the data interlinking task, probably due to the difficulties of matching large collections of data: ObjectCoref and RiMOM. The results of these systems are shown in Figure 8.

The results are very different for two systems, with ObjectCoref being better in precision and RiMOM being better in recall. A difficult task with real interlinked data is to understand if the results are due to a weakness of the matching system or because links can be not very reliable. In any case, what we can conclude from this experiment with linked data is that a lot of work is still required in three directions: i) providing a reliable mechanism for systems' evaluation; ii) improving the performances of matching systems in terms of both precision and recall; iii) work on the scalability of matching techniques in order to make affordable the task of matching large collections of real data. Starting from these challenges, data interlinking will be one one the most important future directions for the instance matching evaluation initiative.

7.2 OWL data task (IIMB & PR)

The OWL data task is focused on two main goals:

1. to provide an evaluation dataset for various kinds of data transformations, including value transformations, structural transformations and logical transformations;
2. to cover a wide spectrum of possible techniques and tools.

To this end, we provided two groups of datasets, the ISLab Instance Matching Benchmark (IIMB) and the Person-Restaurants benchmark (PR). In both cases, participants were requested to find the correct correspondences among individuals of the first knowledge base and individuals of the other. An important task here is that some of the transformations require automatic reasoning for finding the expected alignments.

IIMB. IIMB is composed of a set of test cases, each one represented by a set of instances, i.e., an OWL ABox, built from an initial dataset of real linked data extracted from the web. Then, the ABox is automatically modified in several ways by generating a set of new ABoxes, called *test cases*. Each test case is produced by transforming the individual descriptions in the reference ABox in new individual descriptions that are inserted in the test case at hand. The goal of transforming the original individuals is twofold: on one side, we provide a simulated situation where data referring to the same objects are provided in different data sources; on the other side, we generate different datasets with a variable level of data quality and complexity. IIMB provides transformation techniques supporting modifications of data property values, modifications of number and type of properties used for the individual description, and modifications of the individuals classification. The first kind of transformations is called *data value transformation* and it aims at simulating the fact that data expressing the same real object in different data sources may be different because of data errors or because of the usage of different conventional patterns for data representation. The second kind of transformation is called *data structure transformation* and it aims at simulating the fact that the same real object may be described using different properties/attributes in different data sources. Finally, the third kind of transformation, called *data semantic transformation*, simulates the fact that the same real object may be classified in different ways in different data sources.

The 2010 edition of IIMB is a collection of OWL ontologies consisting of 29 concepts, 20 object properties, 12 data properties and thousands of individuals divided into 80 test cases. In fact, in IIMB 2010, we have defined 80 test cases, divided into 4 sets of 20 test cases each. The first three sets are different implementations of data value, data structure and data semantic transformations, respectively, while the fourth set is obtained by combining together the three kinds of transformations. IIMB 2010 is created by extracting data from Freebase, an open knowledge base that contains information about 11 million real objects including movies, books, TV shows, celebrities, locations, companies and more. Data extraction has been performed using the query language JSON together with the Freebase JAVA API[11]. The benchmark has been generated in a small version consisting in 363 individuals and in a large version containing 1416 individuals. In Figures 9 and 10, we report the results over the large version that are quite similar to the small one.

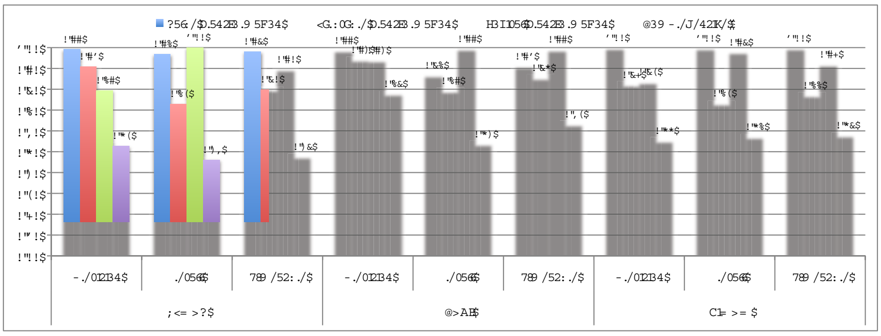


Fig. 9. Results of the IIMB subtrack.

The participation in IIMB was limited to ASMOV, CODI and RiMOM systems. All the systems obtained very good results when dealing with data value transformations and logical transformations, both in terms of precision and in terms of recall. Instead, in case of structural transformations (e.g., property value deletion of addition, property hierarchy modification) and of the combination of different kinds of transformations we have worse results, especially concerning recall. Looking at the results, it seems that the combination of different kinds of heterogeneity in data descriptions is still an open problem for instance matching systems. The three matching systems appear to be comparable in terms of quality of results.

PR. The Person-Restaurants benchmark is composed of three data subsets. Two datasets (Person 1 and Person 2) contain personal data. The Person 1 dataset is created with the help of the Febrl project example datasets[12]. It contains original records of people and modified duplicate records of the same entries. The duplicate record set contains one duplicate per original record, with a maximum of one modification per duplicate record and a maximum of one modification per attribute. Person 2 is created as Person 1, but with a maximum of 3 modifications per attribute, and a maximum of 10 modifications per record. The third dataset (Restaurant) is created with the help of 864 restaurant records from two different data sources (Fodor and Zagat restaurant guides)[13]. Restaurants are described by name, street, city, phone and restaurant category. In all the datasets the number of records is quite limited (about 500/600 entries). Among these, 112 record pairs refer to the same entity, but usually show differences. Results of the evaluation are shown in Figure 11.

The PR task of the instance matching track was quite successful in terms of participation, in that all the five systems sent their results for this task[14]. This is because the PR datasets contain a small number of instances to be matched, resulting in a matching task

---

[11] `http://code.google.com/p/freebase-java/`

[12] Downloaded from `http://sourceforge.net/projects/febrl/`

[13] They can be downloaded from `http://userweb.cs.utexas.edu/users/ml/riddle/data.html`

[14] ASMOV sent a second set of results referred as ASMOV_D. They are the same as ASMOV but alignments are generated using the descriptions available in the TBOX

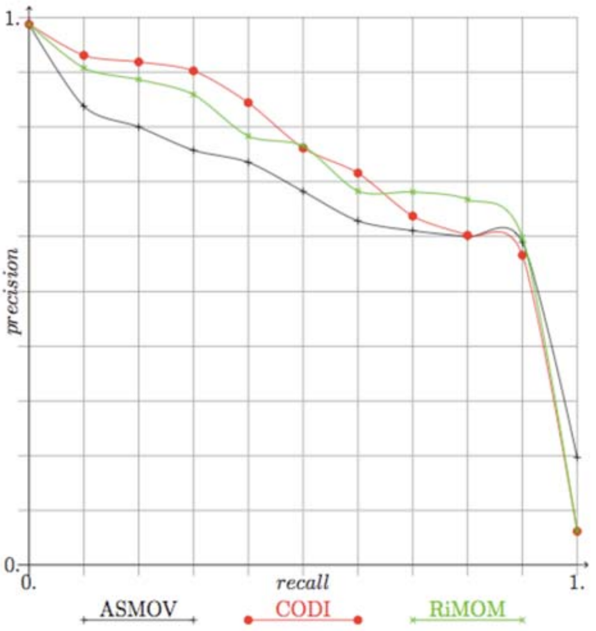


Fig. 10. Precision/recall of tools participating in the IIMB subtrack.

that is affordable in terms of time required for comparisons. The results are good for all systems with the best performances obtained by RiMOM followed by ObjectCoref and LN2R. ASMOV and CODI instead have lower F-measure values in the case of the Person 2 dataset. This is mainly due to low performances in terms of recall. These low recall values depend on the fact that in Person 2 more than one matching counterpart was expected for each person record in the reference dataset.

8 Lesson learned and suggestions

We have seriously implemented the promises of last year with the provision of the first automated service for evaluating ontology matching, the SEALS evaluation service, which has been used for three different data sets. We will continue on this path. We also took into account two other lessons: having rules for submitting data sets and rules for declaring them unfruitful that are published on the OAEI web site. There still remain one lesson not really taken into account that we identify with an asterisk (*) and that we will tackle next year.

The main lessons from this year are:

A) We were not sure that switching to an automated evaluation would preserve the success of OAEI, given that the effort of implementing a web service interface was required from participants. However, we still have as many participants as last year, so this is a good sign.

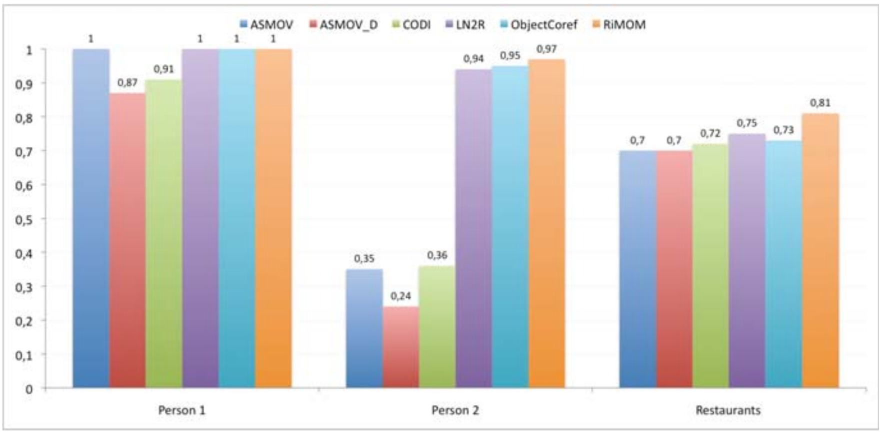


Fig. 11. Results of tools participating in the PR subtrack in terms of F–measure.

B) Although some tools were registered in the SEALS portal, these have not used the SEALS evaluation service either for testing their tools or for registering their final results. We contacted these developers, who answered that they did not have enough time for preparing their tools. So, the effort required for implementing the web service interface and fixing networks issues, has indeed been an obstacle for some participants. However, this effort is low with respect to that required for developing a serious matcher.

C) The SEALS service eases the evaluation execution on a short period because participants can improve their systems and obtain results in real time. This is to some degree also possible for a blind evaluation. This is very valuable.

D) 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 specialized topics. There can be two explanations: the field is shrinking or the entry ticket is too high. To address the first issue we have identified in [17] the challenges in the field to direct research into the critical path.

E) We still can confirm that systems that enter the campaign for several times tend to improve over years. We can also remark that they continue to improve (on data sets in which there is still a progress margin).

F*) The benchmark test case is not discriminant enough between systems. Next year, we plan to introduce controlled automatic test generation in the SEALS evaluation service and think that this will improve the situation.

G) Not all systems followed the general rule to use the same set of parameters in all tracks. In addition, there are systems participating only in one track for which they are specialized. A fair comparison of general-purpose systems, specialized systems and optimally configured systems might require to rethink the application of this rule.

9 Future plans

There are several plans for improving OAEI. The first ones are related to the development of the SEALS service. In the current setting, runtime and memory consumption cannot be correctly measured because a controlled execution environment is missing. Further versions of the SEALS evaluation service will include the deployment of tools in such a controlled environment. As initially planned for last year, we plan to supplement the benchmark test with an automatically generated benchmark that would be more challenging for participants. We also plan to generalize the use of the platform to other data sets. Finally, we would like to have again a data set which requires alignments containing other relations than equivalence.

10 Conclusions

Confirming the trend of previous years, 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).

The trend of number of tracks entered by participants went down again: 2.6 against 3.25 in 2009, 3.84 in 2008 and 2.94 in 2007. This figure of around 3 out of 8 may be the result of the specialization of systems. While, it is not the result of the short time allowed to the campaign, since the SEALS evaluation service had more runs than what the participants registered.

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 also warmly thank Laura Hollinck, Véronique Malaisé and Willem van Hage for preparing the vlcr task which has been cancelled.

We are grateful to Martin Ringwald and Terry Hayamizu for providing the reference alignment for the anatomy ontologies and thank Elena Beisswanger for her thorough support on improving the quality of the data set.

We are grateful to Dominique Ritze for participating in the extension of the reference alignments for the conference track.

We thank Jan Noessner for providing data in the process of constructing the IIMB dataset and we thank Heiko Stoermer and Nachiket Vaidya for providing the PR dataset for Instance Matching.

We also thank the other members of the Ontology Alignment Evaluation Initiative Steering committee: Yannis Kalfoglou (Ricoh laboratories, UK), Miklos Nagy (The Open University (UK), Natasha Noy (Stanford University, USA), Yuzhong Qu (Southeast University, CN), York Sure (Leibniz Gemeinschaft, DE), Jie Tang (Tsinghua University, CN), George Vouros (University of the Aegean, GR).

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

Ondřej Šváb-Zamazal and Vojtěch Svátek were supported by the CSF grant P202/10/1825 (PatOMat project).

References

1. Benhamin Ashpole, Marc Ehrig, Jérôme Euzenat, and Heiner Stuckenschmidt, editors. *Integrating Ontologies'05, Proc. of the K-Cap Workshop on Integrating Ontologies*, Banff (Canada), 2005.
2. Oliver Bodenreide, Terry Hayamizu, Martin Ringwald, Sherri De Coronado, and Songmao Zhang. Of mice and men: Aligning mouse and human anatomies. In *Proc. of the American Medical Informatics Association (AIMA) Annual Symposium*, pages 61–65, 2005.
3. 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. of the 3rd International Workshop on Ontology Matching (OM-2008), collocated with ISWC-2008*, pages 73–120, Karlsruhe (Germany), 2008.
4. Marc Ehrig and Jérôme Euzenat. Relaxed precision and recall for ontology matching. In *Proc. of the K-Cap Workshop on Integrating Ontologies*, pages 25–32, Banff (Canada), 2005.
5. Jérôme Euzenat. An API for ontology alignment. In *Proc. of the 3rd International Semantic Web Conference (ISWC-2004)*, pages 698–712, Hiroshima (Japan), 2004.
6. Jérôme Euzenat, Alfio Ferrara, Laura Hollink, Antoine Isaac, Cliff Joslyn, Véronique Malaisé, Christian Meilicke, Andriy Nikolov, Juan Pane, Marta Sabou, François Scharffe, Pavel Shvaiko, Vassilis Spiliopoulos, Heiner Stuckenschmidt, Ondrej Sváb-Zamazal, Vojtech Svátek, Cássia Trojahn dos Santos, George Vouros, and Shenghui Wang. Results of the ontology alignment evaluation initiative 2009. In *Proc. of the 4th Workshop on Ontology Matching (OM-2009), collocated with ISWC-2000*, pages 73–126, Chantilly (USA), 2009.
7. 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. of the 2nd International Workshop on Ontology Matching (OM-2008), collocated with ISWC-2007*, pages 96–132, Busan (Korea), 2007.
8. 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. of the 1st International Workshop on Ontology*

Matching (OM-2006), collocated with ISWC-2006, pages 73–95, Athens, Georgia (USA), 2006.
9. Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, Heidelberg (DE), 2007.
10. 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.
11. Oktie Hassanzadeh, Reynold Xin, Renée J. Miller, Anastasios Kementsietsidis, Lipyeow Lim, and Min Wang. Linkage query writer. *PVLDB*, 2(2):1590–1593, 2009.
12. Yves R. Jean-Mary, E. Patrick Shironoshitaa, and Mansur R. Kabuka. Ontology matching with semantic verification. *Journal of Web Semantics*, 7(3):235–251, 2009.
13. Anja Jentzsch, Jun Zhao, Oktie Hassanzadeh, Kei-Hoi Cheung, Matthias Samwald, and Bo Andersson. Linking open drug data. In *Proc. of Linking Open Data Triplification Challenge at the I-Semantics*, 2009.
14. Christian Meilicke and Heiner Stuckenschmidt. Incoherence as a basis for measuring the quality of ontology mappings. In *Proc. of the 3rd Workshop on Ontology Matching (OM-2008), collocated ISWC-2008*, pages 1–12, Karlsruhe (Germany), 2008.
15. Vojtěch Svátek Ondřej Šváb-Zamazal O. Empirical knowledge discovery over ontology matching results. In *Proc. of the 1st International Workshop on Inductive Reasoning and Machine Learning on the Semantic Web (IRMLeS-2009), collocated with ESWC-2009*, pages 1–15, Heraklion (Greece), 2009.
16. François Scharffe. *Correspondence Patterns Representation*. PhD thesis, University of Innsbruck, 2009.
17. Pavel Shvaiko and Jérôme Euzenat. Ten challenges for ontology matching. In *Proc. of the 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE-2008)*, pages 1164–1182, Monterrey (Mexico), 2008.
18. York Sure, Oscar Corcho, Jérôme Euzenat, and Todd Hughes, editors. *Proc. of the Workshop on Evaluation of Ontology-based Tools (EON-2004), collocated with ISWC-2004*, Hiroshima (Japan), 2004.
19. Cássia Trojahn dos Santos, Christian Meilicke, Jérôme Euzenat, and Heiner Stuckenschmidt. Automating OAEI campaigns (first report). In *Proc. of the 1st International Workshop on Evaluation of Semantic Technologies (iWEST-2010), collocated with ISWC-2010*, Shanghai (China), 2010.
20. Willem Robert van Hage, Antoine Isaac, and Zharko Aleksovski. Sample evaluation of ontology-matching systems. In *Proc. of the 5th International Workshop on Evaluation of Ontologies and Ontology-based Tools (EON-2007), collocated with ISWC-2007*, pages 41–50, Busan (Korea), 2007.
21. Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Discovering and maintaining links on the web of data. In *Proc. of the 8th International Semantic Web Conference (ISWC-2009)*, pages 650–665, Chantilly (USA), 2009.
22. Ondřej Šváb, Vojtěch Svátek, and Heiner Stuckenschmidt. A study in empirical and 'casuistic' analysis of ontology mapping results. In *Proc. of the 4th European Semantic Web Conference (ESWC-2007)*, pages 655–669, Innsbruck (Austria), 2007.

Grenoble, Milano, Mannheim, Milton-Keynes, Trento, Prague, December 2010

Using AgreementMaker to Align Ontologies for OAEI 2010*

Isabel F. Cruz, Cosmin Stroe, Michele Caci, Federico Caimi, Matteo Palmonari**,
Flavio Palandri Antonelli, Ulas C. Keles

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

Abstract. The AgreementMaker system is unique in that it features a powerful user interface, a flexible and extensible architecture, an integrated evaluation engine that relies on inherent quality measures, and semi-automatic and automatic methods. This paper describes the participation of AgreementMaker in the 2010 OAEI competition in three tracks: benchmarks, anatomy, and conference. After its successful participation in 2009, where it ranked first in the conference track, second in the anatomy track, and obtained good results in the benchmarks track, the goal in this year's participation is to increase the values of precision, recall, and F-measure for each of those tracks.

1 Presentation of the system

We have been developing the AgreementMaker system since 2001, with a focus on real-world applications [5, 8] and in particular on geospatial applications [4, 6, 7, 9–13]. However, the current version of AgreementMaker, whose development started two years ago, represents a whole new effort.

1.1 State, purpose, general statement

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

---

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

** Additional affiliation: University of Milan-Bicocca, Italy.

drives all the matching methods and evaluation strategies; (8) a feedback loop that accepts suggestions and corrections by users and extrapolates new mappings.

In 2009 AgreementMaker was very successful in the OAEI competition. In particular, AgreementMaker ranked (a close) second among ten systems in the anatomy track. AgreementMaker also participated successfully in two other tracks: benchmarks and conference. In the former track, AgreementMaker was ranked first in terms of precision and seventh in terms of recall among thirteen systems and in the latter track AgreementMaker was ranked first with the highest F-measure (57% at a threshold of 75%) among seven competing systems.

1.2 Specific techniques used

AgreementMaker comprises several matching algorithms or *matchers* that can be used for matching (or aligning) the source and target ontologies. The matchers are not restricted to any particular domain. The architecture of AgreementMaker relies on a stack of matchers that belong to three different layers (see Figure 1). Specific configurations of the stack have been used for the benchmarks, anatomy, and conference tracks, as discussed in what follows. However, we describe first the different components in the stack: the matchers, the combination and evaluation modules, and the final alignment module.

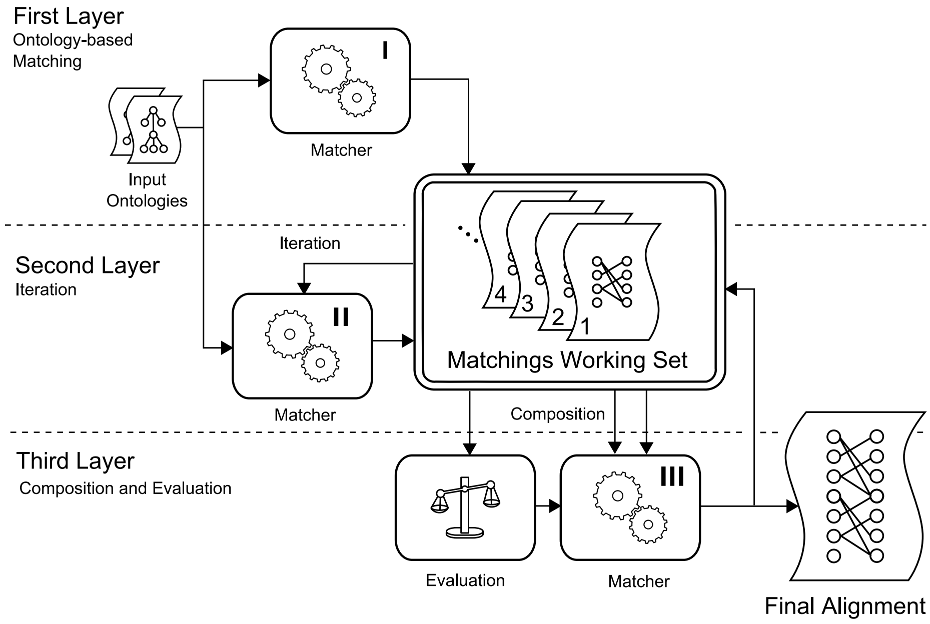


Fig. 1. AgreementMaker OAEI 2010 matcher stack.

Matchers can be concept-based (if they consider only one concept) or structural (if they consider a subgraph of the ontology). The concept-based matchers support the comparison of strings. They include: the Base Similarity Matcher (BSM) [7], the Parametric String-based Matcher (PSM) [2] and the Vector-based Multi-Word Matcher (VMM) [2]. BSM is a basic string matcher that computes the similarity between concepts by comparing all the strings associated with them. PSM is a more in-depth string matcher, which for the competition is set to use a substring measure and an edit distance measure. VMM compiles a *virtual document* for every concept of an ontology, transforms the resulting strings into TF-IDF vectors and then computes their similarity using the cosine similarity measure. These matchers have been extended in the AgreementMaker configuration used this year by plugging in a set of lexicons, which are used to expand the set of strings with synonyms. The extended matchers are therefore called BSM^{lex}, PSM^{lex}, and VMM^{lex}. The Advanced Similarity Matcher (ASM) is a string-based matcher that computes mappings between source and target concepts (including their properties) by comparing their local names, and providing better similarity evaluation in particular when compound terms are used. ASM outperforms generic string-based similarity matchers because it is based on a deeper linguistic analysis.

Structural matchers include the Descendants' Similarity Inheritance (DSI) matcher [7]. This matcher is based on the idea that if two nodes are similar, then their descendants should be similar. The Group Finder Matcher (GFM) is another structural matcher that filters out the mappings provided by another matcher (the input matcher). It identifies groups of concepts and properties in the ontologies and assumes that two concepts (or properties) that belong to two groups that were not mapped by the input matcher will likely have different meanings and should not be mapped. The Iterative Instance Structural Matcher (IISM) takes into account instances. Classes that have mapped individuals can then be aligned. In addition, values of the properties are also considered. The structural part of IISM is quite complex and takes into account superclasses, subclasses, properties, subproperties, cardinalities, and the range and domain of properties.

The combination and evaluation modules are used together, as follows. The Linear Weighted Combination (LWC) [2] combines its inputs (e.g., from several string matchers), using a local confidence quality measure provided by the evaluation module, in order to automatically assign weights to each result computed by the input matchers. After this step, we have a single combined set of alignments that includes the best alignments from each of the input matchers. The final alignment module is given as input a mapping cardinality (e.g., 1:1) and a threshold and outputs the best set of alignments given those two inputs [2].

Benchmarks For the benchmarks track we used the following configuration:

$$\text{IISM(LWC(ASM}, \text{PSM}^{lex}, \text{VMM}^{lex}, \text{BSM}^{lex}) \text{)}$$

LWC is adopted to combine the results of four string-based matchers, namely ASM, PSM^{lex}, PSM^{lex}, and BSM^{lex}; the last three make use of two lexicons, namely WordNet and a dictionary built from the ontologies; the similarity values computed at this step are then given as input to the IISM structural matcher.

Anatomy For the anatomy track we used the following configuration:

$$\text{LWC}(\text{PSM}^{lex}, \text{VMM}^{lex}, \text{BSM}^{lex})$$

LWC is adopted to combine the results of four string-based matchers, namely PSM^{lex}, VMM^{lex}, and BSM^{lex}; the last three make use of two lexicons, namely WordNet and a dictionary built from the ontologies.

Conference For the conference track we used the following configuration:

$$\text{GFM}(\ \text{LWC}(\text{ASM}, \text{PSM})\ )$$

LWC is adopted to combine the results of two string-based matchers, namely ASM and PSM; the similarity values computed at this step are then given as input to the GFM structural matcher.

1.3 Link to the system and parameters file

The AgreementMaker system is available at http://agreementmaker.org/.

1.4 Link to the set of provided alignment (in align format)

The alignment results obtained by AgreementMaker in the OAEI 2010 are available at http://agreementmaker.org/oaei.

2 Results

In this section, we present the results obtained by AgreementMaker in the OAEI 2010 competition. It participated in three tracks: benchmarks, anatomy, and conference. Tests were carried out on a PC running Ubuntu Linux 10.04 with AMD Athlon$^{\text{TM}}$ II X4 635 processor running at 2.9 Ghz and 8 GB RAM.

2.1 Benchmarks

In this track, a source ontology is compared to 111 ontologies that describe the same domain. These ontologies can be divided into 3 categories: concept tests cases (1xx cases), systematic tests cases (2xx cases), and real ontology test cases (3xx cases). AgreementMaker employs the algorithm which we described in section 1.2 for aligning two ontologies.

The 2xx benchmarks test cases are subdivided into 3 groups: 1) 201 to 210, 2) 221 to 247 and 3) 248 to 266. The lexical information in the ontologies in group 1 have been altered to change their labels or identifiers. This alteration includes replacing the labels or identifiers with other names that follow a particular naming convention, a random name, a misspelled name or a foreign word. However, the structure of the ontologies is not modified. The test cases in the second group have ontologies that have flattened hierarchies, expanded hierarchies or no hierarchies at all. The test cases in the third group are the most challenging ones to align. This is because the labels have been scrambled such that they comprise a permutation of letters of a particular length. In addition, the structure of the ontology has been flattened, expanded such that it has more depth or removed completely.

The results obtained by AgreementMaker in the benchmarks track are summarized in Table 1.

	101-104	201-210	221-247	248-266	301-304	H-mean
precision	0.98	0.97	0.95	0.96	0.88	0.95
recall	1.00	0.90	0.99	0.74	0.53	0.79
F-measure	0.99	0.94	0.97	0.82	0.61	0.84

Table 1. Results achieved by AgreementMaker in the benchmarks track of the OAEI 2010 competition.

2.2 Anatomy

This track consists of two real world ontologies to be matched. The source ontology describes the Adult Mouse Anatomy (with 2744 classes) while the target ontology is the NCI Thesaurus describing the Human Anatomy (with 3304 classes). Matching these ontologies is also challenging in terms of efficiency because these ontologies are relatively large. The anatomy track consists of four subtracks: subtrack 1, which emphasizes F-measure, subtrack 2, which emphasizes precision, subtrack 3, which emphasizes recall, and subtrack 4, which tests the capability of extending a partial reference alignment.

The results obtained by AgreementMaker in the anatomy track are summarized in Table 2. We show the precision, recall, and F-measure for subtrack 1, 2 and 3; subtracks 1 and 3 are also evaluated by considering the recall+ measure, which measures how many non trivial correct correspondences, not detectable by string equivalence, can be found in an alignment. Evaluation of subtrack 4 is carried out by analyzing the changes in precision, recall and f-measure when subtrack 1 is compared with subtrack 4.

Anatomy Track	Subtrack 1	Subtrack 2	Subtrack 3	Subtrack 4
precision	0.90	0.96	0.77	+0.03
recall	0.85	0.75	0.87	-0.03
F-measure	0.87	0.84	0.82	+0.00
recall+	0.63	-	0.70	-

Table 2. Results achieved by AgreementMaker in the anatomy track of the OAEI 2010 competition.

2.3 Conference

The conference track consists of 15 ontologies from the conference organization domain and each ontology must be matched against every other ontology. Since the AgreementMaker OAEI 2010 matcher stack considers only two ontologies at a time, we compute 120 alignment files, in total containing 2070 individual alignments. The results obtained are summarized in Table 3. Here we show how precision, recall, and F-measure vary depending on the threshold used for the selection of the mappings.

Threshold	0.60-0.64	0.65-0.69	0.70-0.74	0.75-0.79	0.80-0.84	0.85-0.89	0.90-0.94	0.95-1.00
precision	0.49	0.53	0.61	0.70	0.72	0.75	0.81	0.81
recall	0.64	0.60	0.54	0.52	0.47	0.47	0.46	0.46
F-measure	0.56	0.56	0.58	0.60	0.58	0.58	0.58	0.59

Table 3. Results achieved in the conference track of the OAEI 2010 competition.

2.4 Comments on the results

Benchmarks In the OAEI 2009 competition, AgreementMaker was first in terms of the precision of discovered mappings. However, in terms of recall, AgreementMaker was outperformed by six other systems (thirteen systems competed). The new matchers used in the OAEI 2010 competition address specifically the issue of the alignment of concepts that are not lexically similar. The results of this effort increased the recall by 18% at a cost of 3% in precision in comparison with last year's results. An important contribution to this result comes from the IISM matcher, which exploits instances and structural properties of the ontologies and makes the alignment process less sensitive to lexical differences. A detailed comparison between the results achieved in the 2009 and 2010 competitions in terms of the obtained change in precision, recall, and F-measure for each group of test cases, and the overall H-mean is shown in Table 4.

	101-104	201-210	221-247	248-266	301-304	H-mean
precision 09	0.86	0.73	0.76	0.47	0.83	0.70
precision 10	0.98	0.97	0.95	0.96	0.88	0.95
recall 09	0.86	0.73	0.76	0.47	0.86	0.70
recall 10	1.00	0.90	0.99	0.74	0.53	0.79
F-measure 09	0.92	0.71	0.86	0.45	0.83	0.70
F-measure 10	0.99	0.94	0.97	0.82	0.61	0.84

Table 4. Comparison of the results achieved by AgreementMaker in the 2009 and 2010 OAEI benchmarks track.

Anatomy In comparison with the results achieved by AgreementMaker in the OAEI 2009 competition, the experimental results obtained this year show that the system significantly improved with respect to precision, recall, F-measure, and recall+. A major contribution to these results comes from the exploitation of lexical resources to improve string-based and vector-based matchers. A comparison between the results achieved in the two competitions in terms of precision, recall, F-measure and recall+ for subtracks 1, 2 and 3 is shown in Table 5 (except for recall+ that is not evaluated on subtrack 2). Remarkably, our algorithms for retrieving non trivial mappings significantly improved, as shown by the gain of 0.15 in recall+. Instead, we do not present the comparison with the results obtained in 2009 on subtrack 4, because this year we did not exploit any specific algorithm for propagating mappings available in the input alignment.

Anatomy Track	Subtrack 1	Subtrack 2	Subtrack 3
precision 09	0.87	0.97	0.51
precision 10	0.90	0.96	0.77
recall 09	0.80	0.68	0.82
recall 10	0.85	0.75	0.87
F-measure 09	0.83	0.80	0.63
F-measure 10	0.87	0.84	0.82
recall+ 09	0.49	-	0.55
recall+ 10	0.63	-	0.70

Table 5. Comparison of the results achieved by AgreementMaker in the 2009 and 2010 OAEI anatomy track.

Conference In comparison with the results achieved in OAEI 2009, AgreementMaker significantly improved on precision, recall, and F-measure for thresholds above 0.75 as shown in the graph represented of Figure 2.4, providing more stable results. Remarkably, the new matchers used for the conference track, namely ASM and GFM, can be used on real-world ontologies, since they are based on generic lexical and structural features. Moreover, ASM can be easily adapted to different string-based similarity metrics, and can be extended by adopting a lexicon.

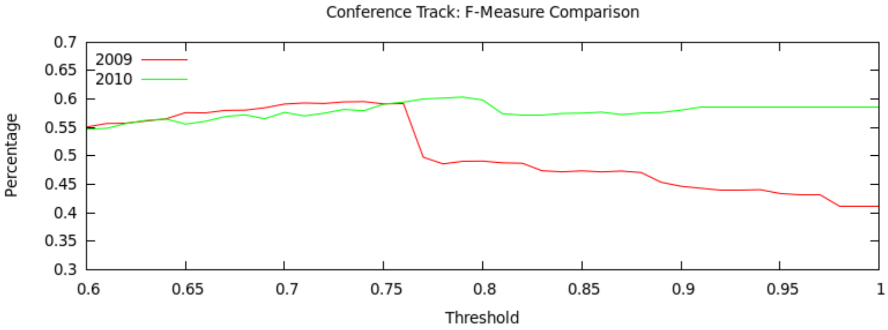


Fig. 2. F-measure comparison for the 2009 and 2010 OAEI conference track results.

3 Conclusions

In this paper we presented the results of the AgreementMaker system for aligning ontologies in the OAEI 2010 competition in the three tracks in which it participated: benchmarks, anatomy, and conference. It was our goal to improve on the results obtained by AgreementMaker in 2009. To meet this goal, we developed several new matching methods, which could be readily integrated into the AgreementMaker system because of its modularity and extensibility. Our results (which we compare with last year's results) amply demonstrate that we have met our goal.

Acknowledgements

Thanks to Catia Pesquita and Francisco M. Couto from Facultade de Ciencias da Universitade de Lisboa, Portugal, for the feedback on the AgreementMaker system, and for useful insights on biomedical ontologies, which significantly increased our understanding of the problem and our results in the anatomy track.

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. 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. 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. 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. 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.

ASMOV: Results for OAEI 2010

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 has consistently been one of the top performing algorithms in the 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 2010 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. There exist many techniques to address this problem [1], differentiated by the way in which different ontology features are exploited. ASMOV, an algorithm that automates the ontology alignment process, uses a weighted average of measurements of similarity along four different features of ontologies, and obtains a pre-alignment based on these measurements. It then uses a unique process of semantic verification to ensure that the alignment does not contain semantic inconsistencies. In this manner, ASMOV was shown to produce the most coherent alignments of all systems tested in OAEI 2009 [3]. A more complete description of ASMOV is presented in [4].

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.

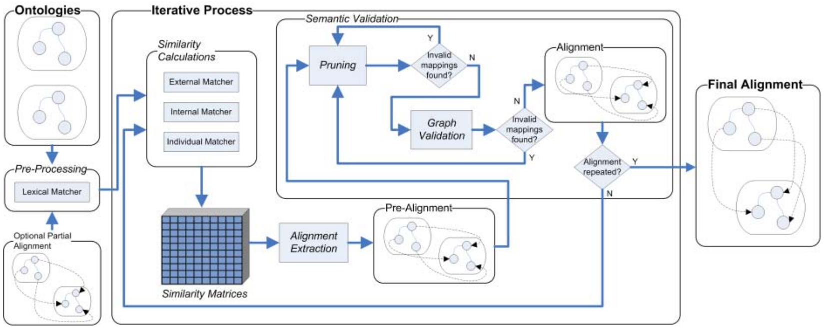


Fig. 1. The ASMOV Mapping Process

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 [5] and ASMOV's ontology modeling component. A thesaurus is optionally used to calculate the lexical similarities between each pair of concepts, properties and individuals. ASMOV can be configured to use either the UMLS Metathesaurus [6] or WordNet [7] in order to derive the similarity measures. If a thesaurus is not used, 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 overall similarity measures (or confidence values) are calculated for each pair. From these similarity measures, a pre-alignment is obtained by selecting the entity from one ontology with the highest similarity for a corresponding entity in the other ontology. A threshold of 0.1% is used to ignore spurious non-zero similarity measures.

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 2009, ASMOV has been improved in three important respects, primarily related to the new instance matching tests. The algorithm in general has been enhanced to allow it to process certain property constructs introduced in OWL 2, especially irereflexive and asymmetric properties. A procedure for disk-based storage of intermediate results has been implemented, allowing the algorithm to handle larger ontologies, although the ontology itself still needs to reside in memory. Further, we have improved the ability of ASMOV to use reasoning enabled by OWL in order to find semantically relevant matches. In particular, we have improved the verification of disjointness between domains and ranges of properties, and we also have included verification of functional properties.

1.3 Adaptations Made for the Evaluation

No special adaptations have been made to the ASMOV system in order to run the 2010 OAEI tests. The stop criterion for ASMOV was established as a multiple-alignment change threshold. For situations where both ontologies have more than 500 concepts, this threshold was set at 1% over three consecutive alignments; otherwise, it was set at 0% over two consecutive alignments. 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://www.infotechsoft.com/products/asmov.aspx.

1.5 Link to the Set of Alignments Produced by ASMOV

The results of the 2010 OAEI campaign for the ASMOV system can be found at http://www.infotechsoft.com/products/asmov.aspx.

2 Results

In this section, we present our comments on the results obtained from the participation of ASMOV in the five tracks of the 2010 Ontology Alignment Evaluation Initiative campaign. 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, with a Web service interface to run with the SEALS environment where required.

2.1 Benchmark

The OAEI 2010 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 these tests in comparison with those obtained in 2009 [8], where ASMOV was found to be one of the two best performing systems [3]. As can be seen, the precision, recall, and F1 measure for the entire suite of tests shows that ASMOV 2010 achieves 98% precision and 89% recall, and an F1 measure of 93%, which represents a 2% improvement over the 2009 version.

The accuracy of ASMOV in the benchmark tests is very high, especially for the 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 10, the most difficult, and with significant improvements at levels 7 through 9 and at the 3xx tests. We believe that these improvements have come specifically through the enhancement of the procedures for utilizing domain and range information in the calculation of internal structure similarity, and through the correction of coding errors and deficiencies. In addition, some of this improvement can be attributable to improvements made in the gold standard.

Table 1. Benchmark test results for ASMOV version 2010 and version 2009

Level	ASMOV 2010			ASMOV 2009		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
0	1.00	1.00	1.00	1.00	1.00	1.00
1	0.99	1.00	0.99	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	0.99	0.98	0.98
4	1.00	0.98	0.99	0.99	0.98	0.98
5	0.99	0.94	0.96	0.97	0.93	0.95
6	0.98	0.90	0.94	0.95	0.89	0.92
7	0.98	0.87	0.92	0.93	0.83	0.88
8	0.98	0.77	0.86	0.90	0.71	0.79
9	0.97	0.64	0.77	0.83	0.48	0.61
10	0.90	0.29	0.44	0.40	0.04	0.07
3xx	0.88	0.84	0.86	0.81	0.82	0.81
All	**0.98**	**0.89**	**0.93**	**0.95**	**0.87**	**0.91**

2.2 Anatomy

For the anatomy track, ASMOV uses the UMLS Metathesaurus [6] instead of WordNet in order to more accurately compute the lexical distance between medical concepts. 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; the SEALS platform provides accuracy measurements for the first three subtasks.

1. *Optimal solution*: The optimal solution alignment is obtained by using the default parameter settings of ASMOV. The accuracy figures obtained from SEALS indicate precision of 79.9% and recall of 77.2%, resulting in overall F1 of 78.5%; these figures are a distinct improvement over the results obtained in 2009.
2. *Optimal precision*: The alignment with optimal precision is obtained by changing the threshold for valid mappings from 0.1% to 30%. The result is that precision increases to 86.5%, while recall decreases to 75.7%. F1 measure is 80.8%, which is higher than our optimal solution, indicating that the use of a higher threshold for ASMOV should be studied more closely.
3. *Optimal recall*: To improve recall, this time ASMOV made use of the annotation property `hasRelatedSynonym` included in the ontologies, to signify synonyms. It should be emphasized that this property is not included in the optimal solution because annotation properties do not have established semantics in OWL, and therefore it would not be possible for a computer to automatically understand that this property actually lists synonyms The results from SEALS indicate that ASMOV found a total of 1521 alignments with precision of 71.7% and recall of 79.2%, resulting in F1 of 75.3%.
4. *Extended solution*: With a partial alignment given as input, the resulting alignment contained all mappings in the partial plus an additional 480 mappings.

2.3 Conference

This collection of tests dealing with conference organization contains 16 ontologies, of which at least one contains constructs specific to OWL 2. ASMOV is able to generate all 120 potential alignments from those ontologies.

Our analysis of the preliminary results obtained in running ASMOV against these ontologies showed a large number of erroneous matches due to incompleteness in our processing of disjointness between property domains and ranges. Specifically, our

Table 2. Results for Conference Test

F1	cmt	confer.	Confof	edas	ekaw	iasted	sigkdd
cmt		0.476	0.378	0.556	0.437	0.364	0.541
confer.			0.718	0.453	0.451	0.286	0.500
confof				0.549	0.681	0.378	0.357
edas					0.529	0.386	0.510
ekaw						0.348	0.368
iasted							0.481

previous versions of ASMOV only verified whether disjointness axioms existed in the asserted domain and range classes. We have now expanded ASMOV to verify any inferred disjointness between domains or ranges based on asserted disjointness within the subsumption hierarchy. Table 2 shows the results that were obtained by running the test through the SEALS platform.

2.4 Directory

Both the "small task" and the "single task" modalities were completed using ASMOV. The "small task" modality comprised 4639 tasks. We compared the results obtained this year against those obtained in 2009, where ASMOV was the best performing system [3]. We found a large degree of agreement, measured as 97% F1. We believe that the difference should result in improvement in the performance of ASMOV over 2009. The "single task" modality consisted of the alignment of a source ontology with 2854 classes, against a target ontology with 6555 classes. ASMOV found a total of 3347 mappings, with a large number of source ontology classes mapped to multiple target ontology classes.

2.5 Instance Matching

The application of ASMOV to the new set of IIMB instance matching tests results in precision of 86%, recall of 82%, and F1 measure of 84% for the small test, and precision of 85%, recall of 80%, and F1 measure of 82% for the large test.

The results of running the persons and restaurants (PR) tests in the SEALS platform are shown in Table 3. We noted the following issues:

- Some conflicts exist between URIs in the TBox (the description ontology) and the ABox. For example, the namespace URI for ontology_people1.owl in the person1 test was http://www.okkam.org/ontology_**people1**.owl in the TBox but http://www.okkam.org/ontology_**person1**.owl in the ABox. We manually corrected the TBox file where these differences were found to enable retrieval of the descriptions of the classes and properties used in the ABox.
- The gold standards for these tests only contained instances of the class "Person" in the person1 and person2 tests, and of the class "Restaurant" in the restaurant test. Running ASMOV in standard fashion produces alignments of instances of other classes such as "Address"; therefore, we restricted ASMOV to only find alignments of a pre-specified class in each test.
- The gold standard also contains mappings between instances that only match in one specific property, when other potential mappings contain matches in more properties. For example, in the Restaurant test, some matches in the gold standard are done exclusively over the "name" property, even if addresses and other property values are different.
- The gold standard also contains mappings between instances that have different values for functional properties. For example, the "surname" property is declared as functional for the class "Person" in the TBox, but two instances with "surname" property "carter" and "carcer" respectively are aligned in the gold

standard. The semantics of functional properties do not allow such an alignment, and ASMOV therefore rejects it. To test the effect of this, we ran ASMOV against the PR tests using and ignoring the TBox in the description ontology. As can be seen in Table 3, ASMOV obtains better results by ignoring the TBox in both Person tests.

Table 3. PR Instance Matching Results

	using TBox		ignoring TBox	
Person1	1.000	0.766	1.000	1.000
Person2	0.982	0.135	0.701	0.235
Restaurants	0.696	0.696	0.696	0.696

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 in the benchmark tests. This is significant since the results in 2009 were already very high. The larger improvements have been obtained in the most difficult tests, showing the utility of the OAEI benchmarks in driving improvement of alignment algorithms. We have also been able to improve our accuracy in the benchmark, directory, and conference tasks. In the instance matching task we find some differences of interpretation with respect to the gold standard, specifically in terms of the semantics of certain properties.

3.2 Discussions on the Way to Improve ASMOV

ASMOV still needs to improve its ability to work with very large ontologies and resources. While some disk-based storage of partial results has been implemented, the entire contents of the ontologies still needs to loaded in memory prior to performing the matching process. This needs to be further improved to use permanent storage in order to enable the alignment of very large ontologies. We also need to continue the implementation of the ability to infer assertions in order to utilize them for similarity measurement and semantic verification. 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. Finally, we need to reexamine the use of an appropriate threshold value to optimize accuracy.

3.3 Comments on the OAEI 2010 Test Cases

The new tests added to the OAEI 2010 contests provide important and welcome tools for the improvement of ontology matching systems. Most importantly, the instance matching task has been made significantly more challenging, allowing us to further refine and expand ASMOV to handle such alignments. Moreover, the availability of an ontology in OWL 2 has allowed us to test some of the improvements made to ASMOV in light of the new standard. In addition, the ability to check accuracy using the SEALS system promises to help significantly in the debugging of our algorithms, once the technical problems with SEALS are resolved. Finally, 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.

One significant problem we found was the extended downtime encountered with the SEALS system. While it is understandable that some technical issues would be encountered, since this is its first deployment for OAEI, we found that SEALS hindered rather than helped in the process of debugging our algorithm and preparing our results. We trust and expect that many of these problems be resolved in the future, as SEALS promises to be a very useful tool for algorithm evaluation.

4 Conclusion

We have presented a brief description of an automated alignment tool named ASMOV, analyzed its performance at the 2010 Ontology Alignment Evaluation Initiative campaign, and compared it with its 2009 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 Dayana Paez, 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. Euzenat J, et.al. Results of the Ontology Alignment Evaluation Initiative 2009. http://www.dit.unitn.it/~p2p/OM-2009/oaei09_paper0.pdf. Accessed 1 Oct 2010.
4. Jean-Mary Y., Shironoshita E.P., Kabuka, M. Ontology Matching with Semantic Verification. Journal of Web Semantics. http://dx.doi.org/10.1016/j.websem.2009.04.001.
5. Jena from HP Labs http://www.hpl.hp.com/semweb/
6. Unified Medical Language System (UMLS) http://umlsks.nlm.nih.gov/
7. WordNet http://wordnet.princeton.edu/
8. Jean-Mary Y, Kabuka M. ASMOV: Results for OAEI 2009. http://www.dit.unitn.it/~p2p/OM-2009/oaei09_paper4.pdf.

BLOOMS on AgreementMaker: results for OAEI 2010

Catia Pesquita[1], Cosmin Stroe [2], Isabel F. Cruz[2], Francisco M. Couto[1]

[1] Faculdade de Ciencias da Universidade de Lisboa, Portugal
cpesquita-at-xldb.di.fc.ul.pt, fcouto-at-di.fc.ul.pt
[2] ADVIS Lab, Department of Computer Science, University of Illinois at Chicago
cstroe1@cs.uic.edu, ifc@cs.uic.edu

Abstract. BLOOMS is an ontology matching method developed as part of an ontology extension system for biomedical ontologies. It combines two lexical similarity measures with similarity propagation. These matchers are applied sequentially, following their precision yield: first lexical similarity based on exact matches, followed by partial matches, and finally these similarities are propagated throughout the ontologies. Partial matches are based on the specificity of words within the ontologies vocabularies. Semantic propagation of similarities is made according to the semantic distance between ontology concepts given by semantic similarity measures. Alignments are extracted after each matcher, to favor precision, since BLOOMS was specifically designed to be as automated as possible. For the participation in OAEI 2010 BLOOMS was integrated into the AgreementMaker system, which provided ontology loading and navigation capabilities. We participated only in the anatomy track, in the tasks #1 and #2 (f-measure and precision), given that BLOOMS was specifically designed for the automated matching of biomedical ontologies. We obtained encouraging results with an f-measure of 0.828 for task #1 and a precision of 0.967 for task #2. Although the current implementation of BLOOMS results in very good precision values, recall is below that of the highest performing systems. This motivates our future work in improving our semantic propagation algorithm and exploiting external resources.

1 Presentation of the system

BLOOMS is an ontology matching method specifically intended for application to biomedical ontologies. The matching of biomedical ontologies has become a focus of interest in recent years due to the increasingly important role that biomedical ontologies are playing in the knowledge revolution that has swept the Life Sciences domain in the last decade. The pressing need for these resources resulted in the parallel development of ontologies by different groups and institutions, giving rise not only to different ontologies covering the same domain, but also to a lack of shared standards and logical links between related ontologies. The alignment of biomedical ontologies is thus crucial to take full advantage of them.

Biomedical ontologies present specific challenges and opportunities for their alignment. One relevant feature of many biomedical ontologies that hinders their alignment is their size, for instance the Gene Ontology contains over 30,000 concepts and ChEBI over 500,000. Many of the systems developed for other domains have difficulty in handling such large ontologies. On the other hand, most biomedical ontologies support few types of relationships, which can hinder the performance of matchers that explore more complex structures. Also, in most biomedical ontologies edges do not all represent the same semantic distance between concepts, for instance, edges deeper in the ontology usually represent shorter distances than edges closer to the root concept.

Another relevant feature is the rich textual information in the form of concept names, synonyms and definitions that most biomedical ontologies have. This can play a crucial role in matching algorithms that exploit lexical resources but it can also be an obstacle since biomedical terminology has a high degree of ambiguity.

In recent years OAEI has been the major play field for biomedical ontologies alignment, in its anatomy track. One important finding of previous OAEI anatomy tracks is that several matches are rather trivial and can be found by simple string comparison techniques. Based on this notion, the work in [1] has applied a simple string matching algorithm to several ontologies available in the NCBO BioPortal, and reported high levels of precision in most cases. There are several possible explanations for this, including the simple structure of most biomedical ontologies, their high number of synonyms and low language variability. To improve on the results of simple string matching, the most successful systems in previous OAEI editions [2,3] have shown the advantages of two distinct strategies: (1) exploitation of external knowledge and (2) composition of different matchers followed by propagation of similarity. The first strategy uses background knowledge resources such as the UMLS to support lexical matching of concepts [4-6]. The second strategy propagates similarities between ontology concepts throughout the ontology graphs, based on the assumption that a match between two concepts should contribute to the match of their adjacent concepts, according to a propagation factor [7].

BLOOMS was designed to leverage on the success of simple lexical matching methods, while still finding alignments where lexical similarity is low, by using global computation techniques. It couples a lexical matching algorithm based on the specificity of words in the ontology vocabulary, with a novel global similarity computation approach that takes into account the semantic variability of edges.

1.1 State, purpose, general statement

The original purpose of BLOOMS is to provide the ontology matching component of an ontology extension system called Auxesia. This system combines ontology matching and ontology learning techniques to propose new concepts and relations to biomedical ontologies. Consequently, BLOOMS was specifically designed to match biomedical ontologies in a fully automated fashion, favoring precision over recall.

Although BLOOMS was specifically designed to be applied to biomedical ontologies, its current implementation is domain-independent since it can function without external forms of knowledge. To capitalize on the specific characteristics of most biomedical ontologies, BLOOMS joins a lexical matcher to exploit the rich textual component with a global similarity computation technique to handle the cases where synonyms exist but are not shared between ontologies. Furthermore, BLOOMS can also exploit annotation corpora, which are available for some biomedical ontologies, to improve the propagation of similarity.

1. Specific techniques used

BLOOMS has a sequential architecture composed of three distinct matchers: Exact, Partial and Semantic Broadcast Match. While the first two matchers are based on lexical similarity, the final one is based on the propagation of previously calculated similarities throughout the ontology graph. Figure 1 depicts the general structure of BLOOMS.

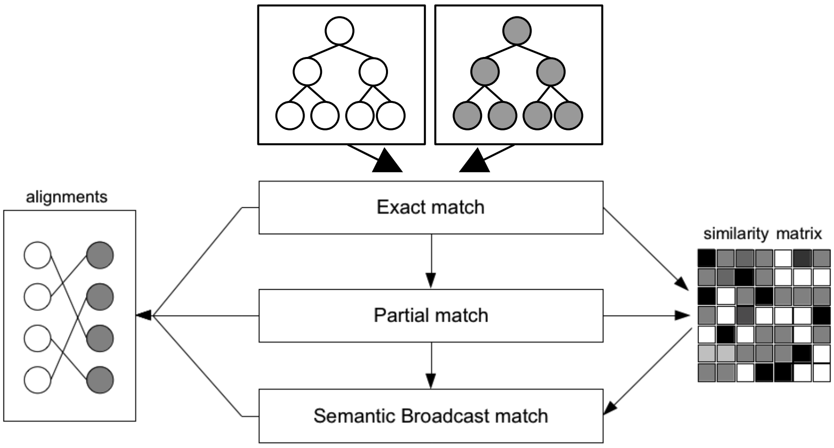


Figure 1. Diagram of BLOOMS architecture. Given two ontologies, BLOOMS first extracts alignments based on Exact matches, then on Partial matches, and finally it propagates the similarities generated by those two strategies using the Semantic Broadcast approach.

1.2 .1 Lexical similarity

Exact and Partial matchers use lexical similarity based on textual descriptions of ontology concepts. Textual descriptors of concepts include their labels, synonyms and definitions. Since ontology concepts usually have several textual descriptors (e.g., name, synonyms, definitions), the similarity between two ontology concepts is given by the maximum similarity between all possible combinations of descriptors.
The first matcher, Exact Match, is run on textual descriptions after normalization and corresponds to a simple exact match, where the score is either 1 or 0.

The second matcher, Partial Match, is applied after processing all concept's labels, synonyms and definitions through tokenizing strings into words, removing stopwords, performing normalization of diacritics and special characters, and finally stemming (Snowball). If the concepts share some of the words in their descriptors, i.e. are partial matches, the final score is given by a Jaccard similarity, which is calculated by the number of words shared by the two concepts, over the number of words they both have. Alternatively, each word can be weighted by its evidence content.

The notion of evidence content (EC) of a word [1] is based on information theory and can be considered a term relevance measure, since it measures the relevance of a word within the vocabulary of an ontology. It is calculated as the negative logarithm of the relative frequency of a word in the ontology vocabulary:

$$EC(word) = -\log freq\left(word \in V_{\text{ontology}}\right)$$

The ontology vocabulary corresponds to all words in all descriptors of all concepts in the ontology. The final frequency of a word within an ontology corresponds to the number of concepts that contain it in any of their descriptors. This means that a word that appears multiple times in the label, definition or synonyms of a concept is only counted once, preventing bias towards concepts that have many synonyms with very similar word sets. The evidence content of words that are common to both ontologies, is given by the average of their ECs within each ontology.

1.2 .2 Semantic Broadcast

After the lexical similarities are computed, they are used as input for a global similarity computation technique, Semantic Broadcast (SB). This novel approach takes into account that the edges in the ontology graph do not all convey the same semantic distance between concepts.

This strategy is based on the notion that concepts whose relatives are similar should also be similar. A relative of a concept is an ancestor or a descendant whose distance to the concept is smaller than a factor *d*. To the initial similarity between concepts, SB adds the sum of all similarities of the alignments between all relatives weighted by their semantic gap *sG*, to a maximum contribution of a factor *c*. This is given by the following:

$$\mathrm{Sim}_{final}(c_a, c_b) = \mathrm{Sim}_{lex}(c_a, c_b) + c\left(\sum \mathrm{Sim}_{lex}(r_i, r_j) \,.\, \mathrm{sG}(c_a, r_i, c_b, r_j)\right)$$
$$\mid D(r_i, c_a) \wedge D(r_j, c_b) < d \wedge r_i, r_j \in A$$

where c_a and c_b are concepts from ontologies *a* and *b*, and r_i and r_j are relatives of c_a and c_b at a distance *D* smaller than a factor *d* whose match belongs to the set of extracted alignments *A*.

The semantic gap between two matches corresponds to the inverse of the average semantic similarity between the two concepts from each ontology. Several metrics can be used to calculate the similarity between ontology concepts, in particular, measures based on information content have been shown to be successful [2].

In BLOOMS we currently implement three information content based similarity measures: Resnik [3], Lin [4] and a simple semantic difference between each concept's ICs. The information content of an ontology concept is a measure of its specificity in a given corpus. Many biomedical ontologies possess annotation corpora that are suited to this application. Nevertheless, semantic similarity can also be given by simpler methods based on edge distance and depth.

Semantic broadcast can also be applied iteratively, with a new run using the similarity matrix provided by the previous.

1.2.3 Alignment Extraction

Alignment extraction in BLOOMS is sequential. After each matcher is run, alignments are extracted according to a predefined threshold of similarity and cardinality of matches, so that the concepts already aligned are not processed for matchers down the line. Each successive matcher has its own predefined threshold.

1.3 Adaptations made for the evaluation

With the purpose of participating in OAEI, BLOOMS was integrated into the AgreementMaker system [5] due to its extensible and modular architecture. We were particularly interested in benefiting from its ontology loading and navigation capabilities, and its layered architecture that allows for serial composition since our approach combines two matching methods that need to be applied sequentially. Furthermore, we also exploited the visual interface during the optimization process of our matching strategy, since although it is not a requirement for our methods, we found it to be extremely useful, it supports a very quick and intuitive evaluation.

Since neither the mouse or the human anatomy ontologies have an annotation corpus, the Semantic Broadcast algorithm used a semantic similarity measure based on edge distance and depth, where similarity decreases with the number of edges between two concepts, and edges further away from the root correspond to higher levels of similarity.

2 Results

BLOOMS was only submitted to the anatomy track, since it is being specifically developed to handle biomedical ontologies. The anatomy track contains 4 tasks: in the first three tasks, matchers should be optimized to favor f-measure, precision and recall, in turn. In the fourth task, an initial set of alignments is given, that can be used to improve the matchers performance. In addition to the classical measures of precision, recall and f-measure, the OAEI initiative also employs recall+, which measures the recall of non-trivial matches, since in the anatomy track a large proportion of matches can be achieved suing simple string matching techniques.

We only participated in tasks #1 and #2, since BLOOMS is designed to favor precision.

2.1 anatomy

Taking advantage of the SEALS platform we ran several distinct configurations of BLOOMS, testing different parameters and also analyzing the contribution of each matcher to the final alignment.

We found that after the first matcher is run, the alignments produced have a very high precision (0.98), but the recall is somewhat low (0.63). Each of the following matchers increases recall while slightly decreasing precision, which was expected given the increasing laxity they provide.

We also found that weighting the partial match score using word evidence content did not significantly alter results when compared to the simple Jaccard similarity.

For task #1 we used a Partial Match threshold of 0.9 and a final threshold of 0.4. Semantic Broadcast was run to propagate similarities through ancestors and descendants at a maximum distance of 2, and contribution was set to 0.4. Using the SEALS evaluation platform, we obtained 0.954 precision, 0.731 recall, for a final F-measure of 0.828 and a recall+ of 0.315.

For task #2 we used a Partial Match threshold of 0.9 and did not use Semantic Broadcast. With this strategy, we ensured a higher precision, of 0.967. However, recall was not much lower than the one in task #1, 0.725 , which resulted in a final f-measure of 0.829.

We did not participate in other tasks, since BLOOMS was originally intended to yield a high precision, as it is intended to be run in a fully automated fashion as a part of an ontology extension system.

3 General comments

We find that the SEALS platform is a very valuable tool in improving matching strategies. We find however that the 100 minute time limit might be detrimental to strategies that need to process large external resources.

3.1 Comments on the results

BLOOMS was designed to be as fully automated as possible, so it is more geared towards increased precision than recall. Comparing our results for tasks #1 and #2, they clearly indicate that our semantic broadcast strategy does not represent a very heavy contribution to recall, but that we do capture nearly 10% more matches when using both the Exact and Partial Match strategies, than Exact Match alone. Also our recall+ is not very high, again highlighting the need to expand our strategy to improve recall.

Nevertheless, we find our performance to be comparable to the best systems in 2009, and in 2010 our f-measure in task #1 is 5% lower that the best performing system, whereas in task #2 we are the second best system, with a slight difference of 0.1% in precision. These are encouraging results and we fully intend to participate in future events with an improved version of BLOOMS.

3.2 Discussions on the way to improve the proposed system

We are planning on implementing several strategies for improvement in the near future, some of which were already a part of our initial strategy, but were not yet implemented at the time of OAEI 2010. To improve the lexical similarity matchers, future versions of BLOOMS will take into account spelling variants and mistakes, and we will also investigate the feasibility of using external resources such as UMLS and WordNet to increase the number of synonyms for both terms and words. We feel this would greatly improve the recall of our strategy. Regarding similarity propagation, we will work extensively on improving our semantic broadcast approach, by exploring alternative strategies for the computation of information content independently of an annotation corpus, and thus expand the number of semantic similarity measures that can be used. We will also adapt semantic broadcast to propagate dissimilarity, and decrease the similarity between concepts that might have a high lexical similarity but very distinct neighborhoods.

4 Conclusion

Participating in the anatomy track of OAEI 2010 has given us an opportunity to evaluate a matching algorithm developed with the practical purpose of being used in a semi-automated ontology extension system, Auxesia. Our matching algorithm, BLOOMS, is intended to be as automated as possible, and thus its current implementation favors precision. This was clearly visible in the results we obtained in tasks #1 and #2 of the anatomy track of OAEI 2010, where we obtained high ranking precision values within the top 3, but lower recall.

In future versions of BLOOMS we will implement several strategies designed to improve recall, while minimizing precision loss.

The lessons learned throughout this period will undoubtedly contribute to an improvement of our method.

Acknowledgements

The work performed at University of Lisbon by Catia Pesquita and Francisco M. Couto was supported by the Multiannual Funding Program and the PhD grant SFRH/BD/42481/2007.
The work performed at UIC by Cosmin Stroe and Isabel F. Cruz has been partially sponsored by the National Science Foundation under Awards IIS-0513553 and IIS-0812258

References

1. Ghazvinian, A., Noy, N. , Musen, M. (2009). Creating mappings for ontologies in biomedicine: Simple methods work. In AMIA Annual Symposium (AMIA 2009), 2. 73
2. Caracciolo, C., Hollink, L., Ichise, R., Meilicke, C., Pane, J. , Shvaiko, P. (2008). Results of the Ontology Alignment Evaluation Initiative 2008. The 7th International Semantic Web Conference. 33, 73
3. Ferrara, A., Hollink, L., Isaac, A., Joslyn, C., Meilicke, C., Nikolov, A., Pane, J., Shvaiko, P., Spiliopoulos, V. , Wang, S. (2009). Results of the Ontology Alignment Evaluation Initiative 2009. Fourth International Workshop on Ontology Matching, Washington, DC , 1. 16, 33, 74
4. Zhang, S., Bodenreider, O. (2007). Lessons Learned from Cross-Validating Alignments between Large Anatomical Ontologies. MedInfo, 12, 822--826. 33
5. Lambrix, P., Tan, H. (2006). SAMBO - system for aligning and merging biomedical ontologies. Web Semantics: Science, Services and Agents on the World Wide Web, 4, 196--206. 31, 33
6. Jean-Mary, Y.R., Shironoshita, E.P. , Kabuka, M.R. (2009). Ontology matching with semantic verification. Web Semantics: Science, Services and Agents on the World Wide Web, 7, 235--251. 33
7. Cruz, I.F., Antonelli, F.P., Stroe, C. (2009). AgreementMaker: Efficient Matching for Large Real-World Schemas and Ontologies. PVLDB, 2, 1586- 1589. 82
8. Couto, F., Silva, M. & Coutinho, P. (2005). Finding genomic ontology terms in text using evidence content. BMC Bioinformatics, 6, S21. 57, 64
9. Pesquita, C., Faria, D., Falcão, A.O., Lord, P., Couto, F.M., Bourne, P.E. (2009). Semantic Similarity in Biomedical Ontologies. PLoS Computational Biology, 5
10. Resnik, P. (1998). 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. Lin D (1998) An information-theoretic definition of similarity. Proc. of the 15th International Conference on Machine Learning. San Francisco, CA: Morgan Kaufmann. pp. 296–304.

CODI: Combinatorial Optimization for Data Integration – Results for OAEI 2010

Jan Noessner and Mathias Niepert

KR & KM Research Group
University of Mannheim, Germany
{jan, mathias}@informatik.uni-mannheim.de

Abstract. The problem of linking entities in heterogeneous and decentralized data repositories is the driving force behind the data and knowledge integration effort. In this paper, we describe our probabilistic-logical alignment system CODI (Combinatorial Optimization for Data Integration). The system provides a declarative framework for the alignment of individuals, concepts, and properties of two heterogeneous ontologies. CODI leverages both logical schema information and lexical similarity measures with a well-defined semantics for A-Box and T-Box matching. The alignments are computed by solving corresponding combinatorial optimization problems.

1 Presentation of the system

1.1 State, purpose, general statement

CODI (**C**ombinatorial **O**ptimization for **D**ata **I**ntegration) leverages terminological structure for ontology matching. The current implementation produces mappings between concepts, properties, and individuals including mappings between object and data type properties. The system combines lexical similarity measures with schema information to reduce or completely avoid *incoherence* and *inconsistency* during the alignment process. The system is based on the syntax and semantics of Markov logic [2] and transforms the alignment problem to a maximum-a-posteriori optimization problem.

1.2 Specific techniques used

Markov logic combines first-order logic and undirected probabilistic graphical models [11]. A Markov logic network (MLN) is a set of first-order formulae with weights. Intuitively, the more evidence there is that a formula is true the higher the weight of this formula. It has been proposed as a possible approach to several problems occurring in the context of the semantic web [2]. We have shown that Markov logic provides a suitable framework for ontology matching as it captures both *hard* logical axioms and *soft* uncertain statements about potential correspondences between entities. The probabilistic-logical framework we propose for ontology matching essentially adapts the syntax and semantics of Markov logic. However, we always *type* predicates and we require a strict distinction between *hard* and *soft* formulae as well as *hidden* and *observable* predicates. Given a set of constants (the classes and object properties of the ontologies) and formulae (the axioms holding between the objects and classes), a Markov logic network defines a probability distribution over possible alignments. We refer the reader to [9, 8] for an in-depth discussion of the approach and some computational challenges. For generating the Marcov logic networks we used the approach described in [12].

T-Box Matching Formalization Given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ and an initial a-priori similarity measure σ we apply the following formalization. First, we introduce observable predicates O to model the structure of $\mathcal{O}_1$ and $\mathcal{O}_2$ with respect to both concepts and properties. For the sake of simplicity we use uppercase letters D, E, R to refer to individual concepts and properties in the ontologies and lowercase letters d, e, r to refer to the corresponding constants in C. In particular, we add ground atoms of observable predicates to $\mathcal{F}^h$ for $i \in \{1, 2\}$ according to the following rules[1]:

$$\mathcal{O}_i \models D \sqsubseteq E \mapsto sub_i(d, e)$$
$$\mathcal{O}_i \models D \sqsubseteq \neg E \mapsto dis_i(d, e)$$
$$\mathcal{O}_i \models \exists R.\top \sqsubseteq D \mapsto sub_i^d(r, d)$$
$$\mathcal{O}_i \models \exists R.\top \sqsupseteq D \mapsto sup_i^d(r, d)$$
$$\mathcal{O}_i \models \exists R.\top \sqsubseteq \neg D \mapsto dis_i^d(r, d)$$

The ground atoms of observable predicates are added to the set of hard constraints $\mathcal{F}^h$, forcing them to hold in computed alignments. The hidden predicates m_c and m_p, on the other hand, model the sought-after concept and property correspondences, respectively. Given the state of the observable predicates, we are interested in determining the state of the hidden predicates that maximize the a-posteriori probability of the corresponding possible world. The ground atoms of these hidden predicates are assigned the weights specified by the a-priori similarity σ. The higher this value for a correspondence the more likely the correspondence is correct *a-priori*. Hence, the following ground formulae are added to $\mathcal{F}^s$:

$$(m_c(c, d),\ \ \sigma(C, D)) \qquad \text{if C and D are concepts}$$
$$(m_p(p, r),\ \ \sigma(P, R)) \qquad \text{if P and R are properties}$$

Notice that the distinction between m_c and m_p is required since we use typed predicates and distinguish between the *concept* and *property* type.

Cardinality Constraints A method often applied in real-world scenarios is the selection of a functional one-to-one alignment [1]. Within the ML framework, we can include a set of hard cardinality constraints, restricting the alignment to be functional and one-to-one. In the following we write x, y, z to refer to variables ranging over the appropriately typed constants and omit the universal quantifiers.

$$m_c(x, y) \wedge m_c(x, z) \Rightarrow y = z$$
$$m_c(x, y) \wedge m_c(z, y) \Rightarrow x = z$$

Analogously, the same formulae can be included with hidden predicates m_p, restricting the property alignment to be one-to-one and functional.

---

[1] Due to space considerations the list is incomplete. For instance, predicates modeling range restrictions are not included.

Coherence Constraints Incoherence occurs when axioms in ontologies lead to logical contradictions. Clearly, it is desirable to avoid incoherence during the alignment process. All existing approaches to alignment repair remove correspondences after the computation of the alignment. Within the ML framework we can incorporate incoherence reducing constraints *during* the alignment process for the first time. This is accomplished by adding formulae of the following type to $\mathcal{F}^h$.

$$dis_1(x,x') \wedge sub_2(x,x') \Rightarrow \neg(m_c(x,y) \wedge m_c(x',y'))$$
$$dis_1^d(x,x') \wedge sub_2^d(y,y') \Rightarrow \neg(m_p(x,y) \wedge m_c(x',y'))$$

Stability Constraints Several approaches to schema and ontology matching propagate alignment evidence derived from structural relationships between concepts and properties. These methods leverage the fact that existing evidence for the equivalence of concepts C and D also makes it more likely that, for example, child concepts of C and child concepts of D are equivalent. One such approach to evidence propagation is *similarity flooding* [7]. As a reciprocal idea, the general notion of stability was introduced, expressing that an alignment should not introduce new structural knowledge [5]. The *soft* formula below, for instance, decreases the probability of alignments that map concepts X to Y and X' to Y' if X' subsumes X but Y' does *not* subsume Y.

$$(sub_1(x,x') \wedge \neg sub_2(y,y') \Rightarrow m_c(x,y) \wedge m_c(x',y'),\ w_1)$$
$$(sub_1^d(x,x') \wedge \neg sub_2^d(y,y') \Rightarrow m_p(x,y) \wedge m_c(x',y'),\ w_2)$$

Here, w_1 and w_2 are *negative* real-valued weights, rendering alignments that satisfy the formulae possible but less likely.

The presented list of cardinality, coherence, and stability constraints could be extended by additional soft and hard formulae. Other constraints could, for example, model known correct correspondences or generalize the one-to-one alignment to m-to-n alignments.

A-Box Matching The current instance matching configuration of CODI leverages terminological structure and combines it with lexical similarity measures. The approach is presented in more detail in [10]. It uses one T-Box $\mathcal{T}$ but two different A-Boxes $\mathcal{A}_1 \in \mathcal{O}_1$ and $\mathcal{A}_2 \in \mathcal{O}_2$. In cases with two different T-Boxes the T-Box matching approach is applied as a preprocessing step, merge the two aligned T-Boxes and then use our instance matching algorithm. CODI offers complete conflict elimination meaning that the resulting alignment is always coherent for OWL DL ontologies. This component is based on the work of Meilicke et al. [6]. CODI enforces the instance alignment to be consistent. To this end, we need to introduce observable predicates O to model conflicts, that is, a positive assertion of one instance in one ontology and a negative assertion of the same instance in the other ontology. This is done for both property and concept assertions.

Analogous to the concept and property alignment before, we introduce the hidden predicate m_i representing instance correspondences. Let C be a concept and P be a property of T-Box $\mathcal{T}$. Further, let $A \in \mathcal{A}_1$ and $B \in \mathcal{A}_2$ be individuals in the respective A-Boxes. Then, using a reasoner, ground atoms are added to the set of *hard* constraints

$\mathcal{F}^h$ according to the following rules:

$$\begin{aligned}
\mathcal{T} \cup \mathcal{A}_1 \models C(A) \wedge \mathcal{T} \cup \mathcal{A}_2 \models \neg C(B) &\quad \mapsto \neg m_i(a,b)\\
\mathcal{T} \cup \mathcal{A}_1 \models \neg C(A) \wedge \mathcal{T} \cup \mathcal{A}_2 \models C(B) &\quad \mapsto \neg m_i(a,b)\\
\mathcal{T} \cup \mathcal{A}_1 \models P(A,A') \wedge \mathcal{T} \cup \mathcal{A}_2 \models \neg P(B,B') &\quad \mapsto \neg m_i(a,b) \vee \neg m_i(a',b')\\
\mathcal{T} \cup \mathcal{A}_1 \models \neg P(A,A') \wedge \mathcal{T} \cup \mathcal{A}_2 \models P(B,B') &\quad \mapsto \neg m_i(a,b) \vee \neg m_i(a',b')
\end{aligned}$$

In addition to these formulae we included cardinality constraints analogous to those used in the concept and property matching of Section 1.2. In the instance matching formulation, the a-priori similarity $\sigma_{\mathbf{c}}$ and $\sigma_{\mathbf{p}}$ measures the *normalized overlap* of concept and property assertions, respectively. For more details on these measures, we refer the reader to [10]. The following formulae are added to the set of soft formulae $\mathcal{F}^s$:

$$\begin{aligned}
(m_i(a,b),\ \ \sigma_{\mathbf{c}}(A,B)) &\qquad \text{if A and B are instances}\\
(m_i(a,b) \wedge m_i(c,d),\ \ \sigma_{\mathbf{p}}(A,B,C,D)) &\qquad \text{if A, B, C, and D are instances}
\end{aligned}$$

1.3 Adaptations made for the evaluation

The strength of the system is its modularity allowing the incorporation of different similarity measures. The system can be optimized in two major ways: (a) Inclusion of novel formulae enforcing the logical consistency and (b) the inclusion of additional similarity measures. There is room for improvement since we used a very simple lexical similarity measure based on the Levenshtein distance [4] for our experiments. It is possible to apply different aggregation functions like average or maximum and to include specific properties of an ontology like URIs, labels, and comments.

In all OAEI test cases Algorithm 1 was used for computing the a-priori similarity $\sigma(entity_1, entity_2)$. In the case of concept and property alignments, the a-priori similarity is computed by taking the maximal similarity between the URIs, labels and *OBO to OWL* constructs. In case of instance matching the algorithm goes through all data properties and takes the average of the similarity scores.

1.4 Link to the System and Parameters File

CODI can be downloaded from `http://codi-matcher.googlecode.com`.

1.5 Link to the Set of Provided Alignments

The alignments for the tracks *Benchmark* and *Conference* has been made with the SEALS platform. For *Anatomy*, *IIMB*, and *Restaurant* the alignments can be found at `http://code.google.com/p/codi-matcher/downloads/list`

2 Results

In the following section, we present the results of the CODI system for the individual OAEI tracks. Due to space considerations, we do not explain the different benchmarks in more detail.

Algorithm 1 $\sigma(entity_1, entity_2)$

```
if $entity_1$ and $entity_2$ are either concepts or properties then
    $value \leftarrow 0$
    for all Values $s_1$ of URI, labels, and OBOtoOWL constructs in $entity_1$ do
        for all Values $s_2$ of URI, labels, and OBOtoOWL constructs in $entity_1$ do
            $value \leftarrow Max(value, sim(s_1, s_2))$
        end for
    end for
    return $value$
end if
if $entity_1$ and $entity_2$ are individuals then
    $Map\langle URI, double\rangle\ similarities \leftarrow null$
    for all dataproperties $dp_1$ of $entity_1$ do
        $uri_1 \leftarrow$ URI of $dp_1$
        for all dataproperties $dp_2$ of $entity_2$ do
            if $uri_1$ equals URI of $dp_2$ then
                $value \leftarrow sim(valueofdp_1, valueofdp_2)$
                if $uri_1$ is entailed in $similarities$ then
                    update entry $\langle uri_1, old\_value\rangle$ to $\langle uri_1$, Minimum $(old\_value + value, 1)\rangle$ in $similarities$
                else
                    add new entry pair $\langle uri1, value\rangle$ in $similarities$
                end if
            end if
        end for
    end for
    return (sum of all values in $similarities$)/(length of $similarities$)
end if
```

Benchmark Track While our system's strength is its modularity and adaptability to different ontologies we used the *exact same setting* for all ontology matching tracks. Hence, the performance on the *benchmark* track is rather poor. This is primarily due to the high threshold of 0.85 for the Levenshtein similarity measure that we applied in each of the ontology matching tracks. The results are shown in Table 1.

Table 1. Benchmark results

	1xx	2xx	3xx	Average
Precision	1	0.70	0.92	0.72
Recall	0.99	0.42	0.43	0.44
F_1 score	1	0.49	0.56	0.51

Conference Track On the real-world conference dataset CODI achieves very good results since it employs logical reasoning to avoid incoherences. The execution time is between 2 and 4 minutes per test case[2]. Table 2 summarizes the overall results.

Table 2. Conference results

	Average
Precision	0.87
Recall	0.51
F_1 score	0.64

Anatomy Track The results on the anatomy track are also convincing. The results shown in Table 3 are en par with the 2009 results of state-of-the-art matching applications. The F_1 scores are between 0.79 and 0.73 for all subtasks, even for the two tasks *Focus on Precision* and *Focus on Recall*. Thus, our algorithm achieves satisfiable precision and recall values without sacrifices on the F_1 score. For the last task, where a partial reference alignment was given, we could gain almost 5 % on the F_1 score. This is because incorporating a partial reference alignment in our system is straight-forward. The reference alignment becomes a direct part of the optimization problem, enforcing good correspondences while ruling out contradicting ones. However, since our algorithm uses logical reasoning and has to solve an NP-hard optimization problem, the execution times are quite high[3].

Table 3. Anatomy results

	Focus on F_1 score	Focus on Precision	Focus on Recall	Partial Alignment
Precision	0.954	0.964	0.782	0.969
Recall	0.680	0.663	0.695	0.742
F_1 score	0.794	0.784	0.736	0.840
Execution Time (min)	88	60	157	95

---

[2] All experiments are executed on a Desktop PC with 2 GB RAM and a Intel Core2 Duo 2.4 GHz processor.

[3] This forces us to submit the solutions without the seals platform because of a timeout after 45 minutes.

IIMB Track The instance matching benchmark IIMB consists of 80 transformations divided in four transformation categories containing 20 transformations each. We applied the full A-Box matching functionality described above with a threshold on the a-priori similarity of 0.1. The average execution time on the IIMB small (large) dataset is 2.6 (35.1) minutes. Table 4 summarizes the different results of the CODI system. The values without brackets are the results for the small IIMB dataset and the values in brackets for the large one.

Table 4. IIMB results

Transformations	0-20	21-40	41-60	61-80[4]	overall
Precision	0.99 (0.98)	0.95 (0.94)	0.96 (0.99)	0.86 (0.86)	0.94 (0.95)
Recall	0.93 (0.87)	0.83 (0.79)	0.97 (0.99)	0.54 (0.53)	0.83 (0.80)
F_1 score	0.96 (0.91)	0.88 (0.85)	0.97 (0.99)	0.65 (0.63)	0.87 (0.85)

PR Track For this track consisting of small files about persons and restaurants, we used a simple one to one alignment only based on lexical similarity scores since no significant structural information is available. Thus, the runtime was with less than 5 seconds per test case very short. The results of the CODI system are depicted in Table 5.

Table 5. PR results

	Person1	Person2	Restaurant
Precision	0.87	0.83	0.71
Recall	0.96	0.22	0.72
F_1-score	0.91	0.36	0.72

3 General comments

3.1 Discussions on the way to improve the proposed system

CODI is a very young system and does not yet provide a user interface. Hence, improvements in usability by designing a suitable user interface will be one of the next steps. In case of the quality of the alignments, more sophisticated lexical similarity measures will be tested and integrated. We are also working on novel algorithms solving the optimization problems more efficiently.

3.2 Comments on the OAEI 2010 procedure

The SEALS evaluation campaign is very beneficial since it is the first time that the matchers must have a standardized interface which could possibly be used by everyone.

3.3 Comments on the OAEI 2010 measures

We encorage the organizers to use semantic precision and recall measures as described in [3].

4 Conclusion

CODI performs concept, property, and instance alignments. It combines logical and structural information with a-priori similarity measures in a well-defined way by using the syntax and semantics of Markov logic. The system therefore not only aligns the entities with the highest lexical similarity but also enforces the coherence and consistency of the resulting alignment.

The overall results of the young system are very promising. Especially when considering the fact that there are many optimization possibilities with respect to the lexical similarity measures that have not yet been investigated. The strength of the CODI system is the combination of lexical and structural information and the declarative nature that allows easy experimentation. We will continue the development of the CODI system and hope that our approach inspires other researchers to leverage terminological structure for ontology matching.

References

1. I. Cruz, F. Palandri, Antonelli, and C. Stroe. Efficient selection of mappings and automatic quality-driven combination of matching methods. In *Proceedings of the ISWC 2009 Workshop on Ontology Matching*, 2009.
2. P. Domingos, D. Lowd, S. Kok, H. Poon, M. Richardson, and P. Singla. Just add weights: Markov logic for the semantic web. In *Proceedings of the Workshop on Uncertain Reasoning for the Semantic Web*, pages 1–25, 2008.
3. D. Fleischhacker and H. Stuckenschmidt. A Practical Implementation of Semantic Precision and Recall. In *2010 International Conference on Complex, Intelligent and Software Intensive Systems*, pages 986–991. IEEE, 2010.
4. V. I. Levenshtein. Binary codes capable of correcting deletions and insertions and reversals. *Doklady Akademii Nauk SSSR*, pages 845–848, 1965.
5. C. Meilicke and H. Stuckenschmidt. Analyzing mapping extraction approaches. In *Proceedings of the Workshop on Ontology Matching*, Busan, Korea, 2007.
6. C. Meilicke, A. Tamilin, and H. Stuckenschmidt. Repairing ontology mappings. In *Proceedings of the Conference on Artificial Intelligence*, pages 1408–1413, Vancouver, Canada, 2007.
7. S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings of ICDE*, pages 117–128, 2002.
8. M. Niepert. A Delayed Column Generation Strategy for Exact k-Bounded MAP Inference in Markov Logic Networks. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2010.
9. M. Niepert, C. Meilicke, and H. Stuckenschmidt. A Probabilistic-Logical Framework for Ontology Matching. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.
10. J. Noessner, M. Niepert, C. Meilicke, and H. Stuckenschmidt. Leveraging Terminological Structure for Object Reconciliation. *The Semantic Web: Research and Applications*, pages 334–348, 2010.
11. M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.
12. S. Riedel. Improving the accuracy and efficiency of map inference for markov logic. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 468–475, 2008.

Eff2Match Results for OAEI 2010

Watson Wei Khong Chua[1] and Jung-Jae Kim[2]

[1] Nanyang Technological University, Singapore `watsonchua@pmail.ntu.edu.sg`
[2] Nanyang Technological University, Singapore `jungjae.kim@ntu.edu.sg`

Abstract. While the primary objective of an ontology alignment tool is to identify as many correct correspondences as possible, efficiency in terms of run-time needs to be achieved for practical usage. Not only does run-time efficiency enable scalability, it also facilitates information integration for time-critical applications using heterogeneous ontologies. In this paper, we present our ontology alignment approach known as *Eff2Match* which aligns a pair of ontologies with high accuracy and low runtime.

1 Presentation of the system

Ontologies are being widely used for semantic representation in applications from various domains such as biomedical informatics [3] and earth sciences [5]. They can be used to provide data with semantics, thus resolving the heterogeneity problem between information sources at the data level. However, the problem is only partially resolved because different ontology engineers model their ontologies differently. Therefore, the heterogeneity problem is escalated to the ontological level if information is to be shared between applications using different ontologies. As a result, ontology alignment tools that can achieve high accuracy are required. In this paper, we present *Eff2Match* (pronounced "Eff Squared match"), an **Eff**ective and **Eff**icient ontology matching tool which can match a pair of ontologies with good accuracy within a short amount of time. *Eff2Match* uses an effective and dynamic candidate reduction technique to avoid performing unnecessary comparisons, thereby achieving high efficiency.

1.1 State, purpose, general statement

In order to facilitate the sharing of information among applications using different ontologies, we have developed an automatic ontology alignment tool called *Eff2Match*. It is able to align both concepts and properties in different ontologies that are semantically equivalent (concepts are matched to concepts and properties are matched to properties). The current implementation does not match the instances in the ontologies.

1.2 Specific techniques used

Eff2Match takes as input the URI of a pair of ontologies to be aligned and matches entities (concepts or properties) in the source ontology to those in the target ontology. The alignment process consists of four stages: 1) Anchor Generation, 2) Candidates Generation, 3) Anchor Expansion and 4) Iterative Score Boosting as shown in Fig. 1 .

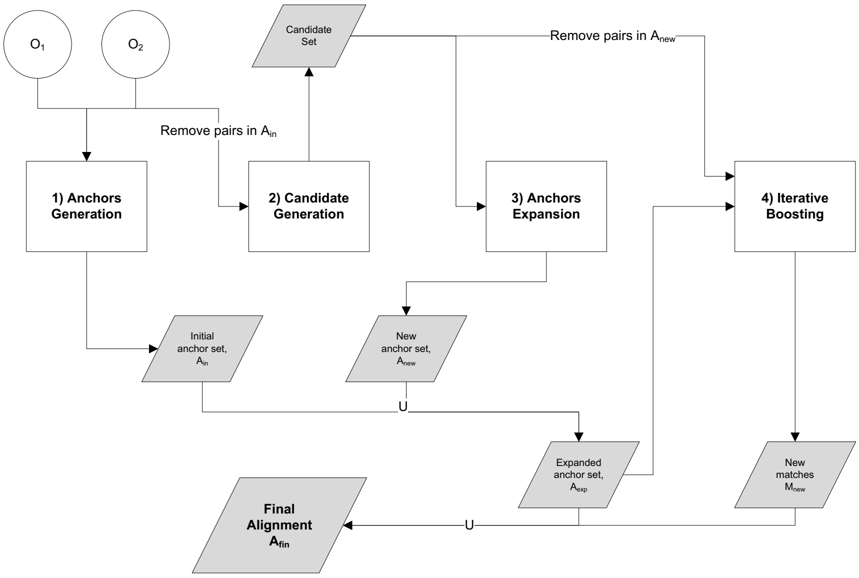


Fig. 1. Eff2Match Algorithm Flow

Anchor Generation In the Anchor Generation stage, matching entities are identified using an exact string matching technique. Local names and labels of entities $e2_j$ in the target ontology are first preprocessed through camel case conversion, case-normalization and removal of delimiters. A hash-table is then used to map the preprocessed local names and labels to their corresponding entities. After that, we preprocess the local name and label for each entity $e1_i$ in the source ontology and look them up in the hash-table. If either a matching local name or label can be found in the hash-table, we consider the corresponding entity $e2_j$ in the target ontology to be equivalent to the source entity. This method is significantly faster than a pairwise comparison of local names and labels as it takes only $O(n_1) + O(n_2)$ time compared to the latter which requires $O(n_1 \times n_2)$ where n_1 and n_2 are the number of entities in the source ontology O_1 and target ontology O_2 respectively.

Candidate Generation In the Candidates Generation stage, we enumerate candidates for entities in the source ontology that has not been matched in the previous stage using a Vector Space Model (VSM) approach. For each concept, we generated three VSM vectors from the annotations (local name, label and comments) in the ancestors (Vec_a), descendants (Vec_d) and the concept (Vec_c) itself. For each property, the vectors generated consist of the annotations in the property (Vec_p) itself, the property's domain concepts (Vec_{do}) and range concepts(Vec_r). The VSM similarity (Sim_{VSM}) between two concepts $c1_i$ and $c2_j$ is an aggregation of the cosine similarity between the concept VSM vectors, ancestor VSM vectors and descendant VSM vectors using a weighted average.

$$Sim_{VSM} = \frac{\alpha \times Vec_{c1i} \cdot Vec_{c2j} + \beta \times Vec_{a2i} \cdot Vec_{a2j} + \gamma \times Vec_{d2i} \cdot Vec_{d2j}}{\alpha + \beta + \gamma}$$

where α, β and γ are the weights given to the similarity between annotations of the concepts themselves, annotations of their ancestor concepts and annotations of their descendant concepts respectively. The similarity values for properties are calculated in a similar manner and two matrices, M_{con} and M_{prop} are used to store the similarity values for concepts and properties respectively. The VSM similarities are normalised to [0,1] by dividing each entry in M_{con} and M_{prop} by their largest value to get the normalised VSM similarity, $Sim_{VSM_N}(C1_i, C2_j)$. Candidates selection is then performed for each source entity by taking the top-K entities in the target ontology according to their VSM similarities.

Anchor Expansion In the anchor expansion stage, more equivalent pairs of entities are identified by comparing the source entities with their candidate entities using terminological methods. In *Eff2Match*, a term-removing algorithm (TRA) is used for efficiency purposes and the algorithm is illustrated in Fig. 2. First, the labels (local names) of a pair of entities are tokenised. The tokens are then stemmed and words that are stemmed to the same form are removed from both labels. If there are tokens remaining in both the labels, the next stage compares tokens from different labels pairwise using WordNet to determine if they are synonyms of each other. Synonymous tokens are then removed from the labels. If there are no tokens remaining in both the labels after any stage, the two entities are considered to be equivalent and added to the anchor set.

If there are remaining tokens in only one of the labels and not the other, we use a novel technique known as Informative Word Matching (IWM) to determine if the entities are matches. If concept $C1_i$ contains p more terms than $C2_j$ and these p terms occur in the labels of the ancestors of $C2_j$, we can consider $C1_i$ and $C2_j$ to have the same meaning. For example, if $C1_i$ has the label *Heart Endocardium* and $C2_j$ has the label *Endocardium* and the $C2_j$ is a sub-concept of *Heart Part*, we can determine that $C1_i$ and $C2_j$ are semantically equivalent from their labels since the word *Heart* in $C1_i$ is not informative.

Given a pair of entities e_{emp} and e_{rem} where e_{emp} is the entity without remaining tokens in its label and e_{rem} is the entity with p remaining tokens in its label, the following steps are performed to determine if the p remaining tokens are informative words:

1. Collect the labels of ancestors of e_{emp} up to r generations or when the root of the ontology is reached, whichever is earlier.
2. Tokenise and stem the collection of labels collected to get a set of stemmed ancestor tokens S_{sat}.
3. For each token t_i, $i \in [1..p]$ remaining in e_{rem}, stem t_i and check if it exists in S_{sat}. If it does, the word is not informative and it is removed it from e_{rem}.
4. Look up the definition of the original label of e_{emp} in WordNet, tokenise and stem the words in the definition before adding them to the set of definition words, S_{def}.

5. For each token t_i, $i \in [1..p]$ remaining in e_{rem}, stem t_i and check if it exists in S_{def}. If it does, the word is not informative and it is removed it from e_{rem}.

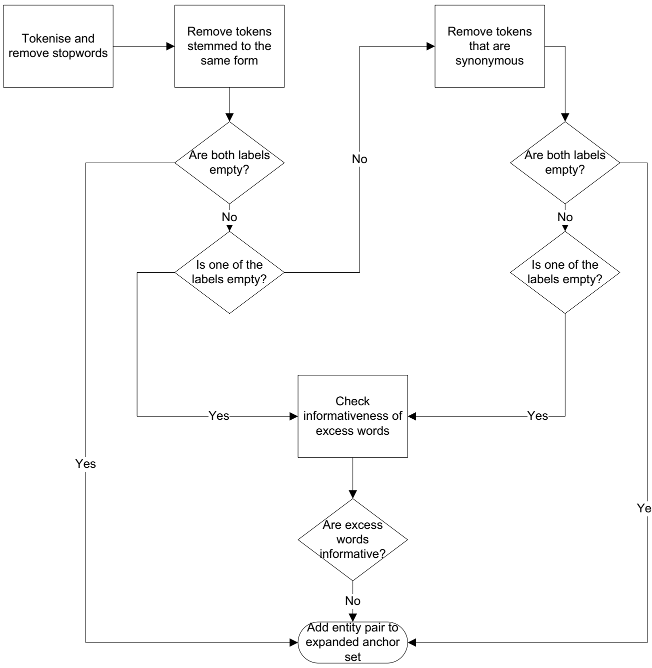


Fig. 2. Anchor Expansion Flow Chart

Iterative Boosting In the final stage of the matching process, an iterative boosting (IterBoost) process is used to identify more pairs of equivalent concepts using the expanded anchor set A_{exp}. In this stage, the algorithm attempts to match the source concepts that have not been matched with their candidates iteratively. In each iteration, the source concepts are ranked based on the sum of their ancestors and descendants with matches in A_{exp}. The top K source concepts are then selected and a formula is used to boost the score of their candidates based on the number of common ancestors and descendants that they share. Given a source concept $C1_i$ and a candidate concept $C2_j$, the structural overlap $SO(C1_i, C2_j)$ between them is calculated using:

$$SO(C1_i, C2_j) = \frac{|\xi(A(C1_i), A(C2_j))| + |\xi(D(C1_i), D(C2_j))|}{min(|A(C1_i)|, |A(C2_j)|) + min(|D(C1_i)|, |D(C2_j)|)}$$

where $A(C)$ is the set of ancestors of concept C and $D(C)$ is the set of descendants of concept C and the function $\xi(X, Y)$ enumerates the set of concepts in X which have equivalences in Y. The equivalences were determined from the comparison in the previous stages. The similarity between $C1_i$ and $C2_j$ is then given by:

$$Sim_{boost}(C1_i, C2_j) = \begin{cases} Sim_{VSM_N}(C1_i, C2_j), & SO(C1_i, C2_j) > t_b \\ \sqrt{Sim_{VSM_N}(C1_i, C2_j)}, & otherwise. \end{cases}$$

The highest scoring candidate is then selected to be the matching concept and the confidence that the pair matches is given by $Sim_{boost}(C1_i, C2_j)$. If $Sim_{boost}(C1_i, C2_j)$ is greater a cut-off threshold t_c, $C1_i$ and $C2_j$ are inserted into A_{exp} as well as the final set of alignment and the process is repeated until all the source entities and their candidates have been visited. If $Sim_{boost}(C1_i, C2_j)$ is less than t_c, $C1_i$ and $C2_j$ are inserted into the set of final alignment but are not considered anchors for future iterations.

1.3 Adaptations made for the evaluation

No special adaptations were made for individual tracks and all alignment processes make use of the same set of parameters. The cut-off threshold for the correspondences was set at 0.7 for the best F-Measure. The only external resource that we used is WordNet. For Informative Word Matching, we set $p = 1$, meaning that we only perform IFM for entities with one remaining token after the Term-Removing Algorithm (TRA). For *IterBoost*, the cut-off threshold for boosting, t_b is set at 0.4 while the cut-off threshold for matching entities to be considered anchors, t_c is set at 0.5. Lastly, the weights α, β, and γ are set to 2, 1, 1, respectively. The matcher has also been implemented as a web service so that it can be evaluated on the SEALS platform.

1.4 Link to the system and parameters file

The Eff2Match system (jar file) and configuration files can be found at [3]
`http://www.cais.ntu.edu.sg/~chua0507/OAEI/Eff2MatchSystem.zip`
with installation instructions at
`http://www.cais.ntu.edu.sg/~chua0507/OAEI/Instructions.txt`

1.5 Link to the set of provided alignments (in align format)

The set of alignments produced by Eff2Match can be found at
`http://www.cais.ntu.edu.sg/~chua0507/OAEI/Eff2MatchAlignments.zip`

2 Results

Eff2Match participated in the benchmark, anatomy and conference tracks of the OAEI 2010 competition and the results are presented in the following subsections:

---

[3] Please replace the ˜symbol by typing it in manually using your keyboard.

2.1 Benchmark

The ontologies in the benchmark dataset can be categorised into 5 different categories according to the difficulty of matching as shown in Table 2.1. *Eff2Match* performs well on all the categories, achieving an F-Measure of more than 0.75 with the exception of the category 248 – 266. Ontologies in this category have different linguistics and structural characteristics, which are core attributes which *Eff2Match* relies on for finding correspondences, thus explaining the poor results.

Ontologies	Precision	Recall	F-Measure
101-104	1.000	1.000	1.000
201-210	0.986	0.684	0.768
221-247	0.990	1.000	0.995
248-266	0.929	0.502	0.591
301-304	0.889	0.711	0.780

Table 1. Results for benchmark dataset

2.2 Anatomy

The anatomy dataset consists of two large real world ontologies, namely the Adult Mouse Anatomy with 2247 classes and the anatomy part of the NCI Thesaurus with 3304 classes. *Eff2Match*'s results for this track are shown in Table 2.2 and shows that *Eff2Match* can match large ontologies with high accuracy and short run-time.

Task	Precision	Recall	F-Measure	Recall+	Time Taken
Task 1 (Optimal F-Measure)	0.955	0.781	0.859	0.440	2.5 mins
Task 2 (Optimal Precision)	0.968	0.745	0.842	N.A.	2.5 mins
Task 3 (Optimal Recall)	0.766	0.838	0.800	0.588	2.5 mins

Table 2. Results for anatomy dataset

2.3 Conference

Lastly, Table 2.3 presents results for *Eff2Match* in the conference track for the 21 ontologies where reference alignments are available. As these ontologies are developed heterogeneously, discovering the correct correspondences between them is more difficult than for the other two tracks. *Eff2Match* was able to achieve F-Measures ranging from 0.4 to 0.759 for pairs of ontologies in this track.

Ontologies	Precision	Recall	F-Measure
cmt-ekaw	0.316	0.545	0.400
conference-edas	0.303	0.588	0.400
conference-iasted	0.350	0.500	0.412
cmt-iasted	0.267	1.000	0.421
edas-iasted	0.471	0.421	0.444
cmt-conference	0.467	0.438	0.452
cmt-confof	0.538	0.438	0.483
conference-ekaw	0.481	0.520	0.500
edas-ekaw	0.500	0.522	0.511
cmt-edas	0.385	0.769	0.513
confof-edas	0.500	0.684	0.578
ekaw-sigkdd	0.538	0.636	0.583
confof-sigkdd	0.667	0.571	0.615
conference-sigkdd	0.588	0.667	0.625
conference-confof	0.550	0.733	0.629
confof-iasted	0.600	0.667	0.632
iasted-sigkdd	0.500	0.867	0.634
ekaw-iasted	0.533	0.800	0.640
edas-sigkdd	0.611	0.733	0.667
confof-ekaw	0.824	0.700	0.757
cmt-sigkdd	0.647	0.917	0.759
Average	0.506	0.653	0.555

Table 3. Results for conference dataset

3 General comments

3.1 Comments on the results

This is the first time that *Eff2Match* is participating in the OAEI competition and it has shown good results compared to the results of other systems in the 2009 competition. In particular, for the anatomy track, its F-Measure of 0.857 tops the best F-Measure achieved in the OAEI 2009 anatomy track. What is more remarkable is that this was achieved with a runtime of only around 2.5 minutes, thereby living up to its name of being an effective and efficient matcher. In addition, its average F-Measure of 0.555 for the conference track ranks second when compared with systems participating in OAEI 2009.

3.2 Discussions on ways to improve *Eff2Match*

The current implementation of *Eff2Match* only matches concepts and properties with equivalence relations. The techniques used are mainly terminological and structural. Our first proposed improvement to *Eff2Match* is to extend its functionalities to include the discovery of non-equivalence correspondences. Other than the subsumption and disjointedness correspondences defined in OWL, *Eff2Match* will also discover other pre-defined relations that are common within the ontologies to be aligned. For example, in the biomedical domain, many ontologies in the OBO foundry [1] contains the *part-of* relationship but they only connect concepts within the same ontology. We intend to discover relationships like these in a future version of *Eff2Match*.

In addition, the current version of *Eff2Match* requires a few parameters to be set by the user. The tuning of parameters is a manual process which can be tedious and ineffective. Our other proposed improvement to *Eff2Match* is to enable it to tune the parameters automatically, like what has been done in [2].

3.3 Comments on the OAEI 2010 procedure, test cases and measures

In this year's OAEI competition, evaluation was done on the SEALS platform [4] for the benchmark, anatomy and conference tracks. Matchers participating in this track have to be implemented as web services so that they can be evaluated on the SEALS platform. We feel that this service is very useful for us, particularly for the anatomy track. Unlike the benchmark and conference tracks where the reference alignments are made known to us, evaluation for the anatomy track is done using a blind test. Therefore, it is not possible for us to observe how changes we make to the algorithm affect the results on the anatomy dataset and it is difficult to make improvements. The SEALS platform has alleviated this problem and made evaluation for the anatomy track possible.

Another useful feature of the SEALS evaluation mechanism is that it shows the correct, incorrect and missing correspondences to the participants, allowing them to gain a greater insight of the strengths and weaknesses of their systems. Though this is an extremely useful feature, we feel that the correspondences for the anatomy dataset should not be shown if it were to remain a blind test. The reason is that by joining the set of correct correspondences and the set of missing correspondences, one can easily get hold of the complete reference alignment for the anatomy dataset.

4 Conclusion

We have presented an ontology matcher named *Eff2Match* that can align ontologies efficiently and effectively. Experiments were performed on different pairs of ontologies from three different tracks in OAEI 2010 and results show that *Eff2Match* is able to match real-world ontologies accurately. In addition, it scales well to large ontologies and can be used in applications where the ontology matching process has to be fast.

References

1. The Open Biomedical Ontologies. Available at: `http://obofoundry.org/about.shtml`.
2. Mayssam Sayyadian, Yoonkyong Lee, AnHai Doan, and Arnon Rosenthal. Tuning schema matching software using synthetic scenarios. In *Proceedings of 31st International Conference on Very Large Data Bases (VLDB'05)*, pages 994–1005, Trondheim, Norway, 2005.
3. Nadine Schuurman and Agnieszka Leszczynski. Ontologies for bioinformatics. *Bioinform Biol Insights*, 2:187–200, 2008.
4. SEALS-Semantic Evaluation At Large Scale. Available at: `http://seals.inrialpes.fr/platform`.
5. Semantic Web for Earth and Environmental Terminology (SWEET). Available at: `http://sweet.jpl.nasa.gov/ontology/`.

ObjectCoref & Falcon-AO: Results for OAEI 2010

Wei Hu, Jianfeng Chen, Gong Cheng, and Yuzhong Qu

[1] Department of Computer Science and Technology, Nanjing University, China
[2] State Key Laboratory for Novel Software Technology, Nanjing University, China
`{whu, yzqu}@nju.edu.cn`

Abstract. In this report, we mainly present an overview of ObjectCoref, which follows a self-training framework to resolve object coreference on the Semantic Web. Besides, we show preliminary results of Falcon-AO (2010) for this year's OAEI campaign, including the benchmark and conference tracks.

1 Presentation of the system

1.1 State, purpose, general statement

The Semantic Web is an ongoing effort by the W3C Semantic Web Activity to actualize data integration and sharing across different applications and organizations. To date, a number of prominent ontologies have emerged to publish data for specific domains, such as the Friend of a Friend (FOAF). These specifications recommend common identifiers for classes and properties in the form of *URIs* [1] that are widely and consistently used across data sources.

On the instance level, however, it is far from achieving agreement among sources on the use of common URIs to identify specific *objects* on the Semantic Web. In fact, due to the decentralized and dynamic nature of the Semantic Web, it frequently happens that different URIs from various sources, more likely originating from different RDF documents, are used to identify the same real-world object, i.e., refer to an identical thing (as known as URI aliases [5]). Examples exist in the domains of people, academic publications, encyclopedic or geographical resources.

Object coreference resolution, also called consolidation or identification [2], is a process for identifying multiple URIs of the same real-world object, that is, determining URI aliases (called coreferent URIs in this report) that denote a unique object. At present, object coreference resolution is recognized to be useful for data-centric applications, e.g. heterogeneous data integration or mining systems, semantic search, query and browsing engines.

We introduce a new approach, *ObjectCoref*, for bootstrapping object coreference resolution on the Semantic Web. The architecture of the proposed approach follows a common self-training framework (see Fig. 1). Self-training [6] is a major kind of semi-supervised learning, which assumes that there are abundant unlabeled examples in the real world, but the number of labeled training examples is limited. We believe that self-training is an appropriate way for resolving object coreference on the Semantic Web.

Falcon-AO [4] is an automatic ontology matching system with acceptable to good performance and a number of remarkable features. It is written in Java, and is open source. ObjectCoref and Falcon-AO together help better enable interoperability between applications that use heterogeneous Semantic Web data.

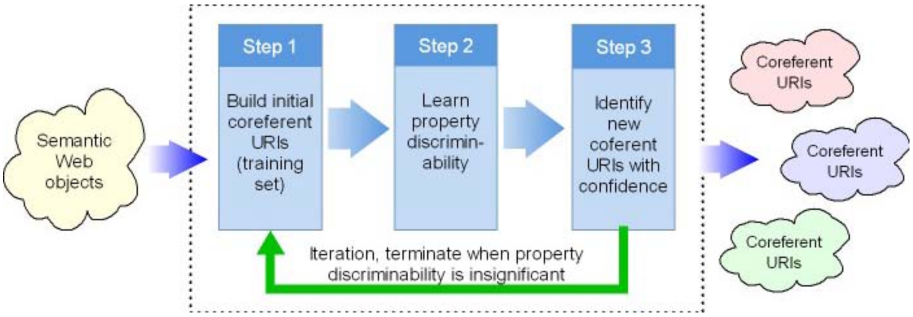


Fig. 1. Self-training process

1.2 Specific techniques used

ObjectCoref builds an initial set of coreferent URIs mandated by the formal and explicit semantics of owl:sameAs, owl:InverseFunctionalProperty, owl:FunctionalProperty, owl:cardinality and owl:maxCardinality.

The semantics of owl:sameAs dictates that all the URIs linked with this property have the same identity; if a property is declared to be inverse functional (IFP), then the object of each property statement uniquely determines the subject (some individual); a functional property (FP) is a property that can have only one unique value for each object; while cardinality (or max-cardinality) allows the specification of exactly (or at most) the number of elements in a relation, in the context of a particular class description, and when the number equals 1, it is somehow similar to the FP, but only applied to this particular class.

Next, ObjectCoref learns the discriminability of pairs of properties based on the coreferent URIs, in order to find more coreferent URIs for extending the training set. The discriminability reflects how well each pair of properties can be used to determine whether two URIs are coreferent or not. As an extreme example, IFPs (e.g. foaf:mbox) have a very good discriminability.

In RDF graphs, each URI is involved in a number of RDF triples whose subject is the URI, and the predicates and objects in these RDF triples form some <property, value> pairs, which can be considered as features for describing such URI. ObjectCoref compares the values between the <property, value> pairs from coreferent URIs, and finds which two properties have similar values and how frequent. The significance is the percentage of the number of coreferent URIs that can found by the discriminant properties in all the coreferent URIs in the training set. If the significance is greater than a given threshold, such the property pair is chosen for further resolution. Please note that for different domains, same property pairs may have different discriminability.

For example, a pair of rdfs:labels is discriminant for the biomedical domain but not for people.

If new coreferent URIs are found, ObjectCoref selects highly accurate ones and adds them into the training set. The whole process iterates several times and terminates when the property discriminability is not significant enough or cannot find more discriminant property pairs.

1.3 Adaptations made for the evaluation

For ObjectCoref, there is no explicit equivalence semantics in the DI and PR tracks. In order to establish the initial training set of coreferent URIs, we randomly extract 20 mappings from the reference alignment for each test case. All the mappings generated by ObjectCoref are based on the same parameters.

For Falcon-AO, we do not make any specific adaptation in the OAEI 2010 campaign. All the mappings for the benchmark and conference tracks outputted by Falcon-AO are uniformly based on the same parameters.

1.4 Link to the system and parameters file

We implement an online service for ObjectCoref, and run it over a large-scale dataset collected by the Falcons [3] search engine up to Sept. 2008. The dataset consists of nearly 600 million RDF triples describing over 76 million URIs. It is still under development. Please visit: `http://ws.nju.edu.cn/objectcoref`.

Besides, we follow the SEALS platform to publish Falcon-AO (2010) as a service. Please access it from `http://219.219.116.154:8083/falconWS?wsdl`. The offline version can be downloaded from our website: `http://ws.nju.edu.cn/falcon-ao`.

1.5 Link to the set of provided alignments (in align format)

The alignments for this year's OAEI campaign should be available at the official website: `http://oaei.ontologymatching.org/2010/`.

2 Results

In this section, we will present the results of ObjectCoref and Falcon-AO (2010) on the tracks provided by the OAEI 2010 campaign.

2.1 DI

In this track, we use ObjectCoref to resolve object coreference between three pairs of datasets, namely diseasome vs. sider, dailymed vs. sider and drugbank vs. sider. Table 1 shows the discriminant property pairs that ObjectCoref learns by self-training. For example, diseasome:name and sider:siderEffectName are a pair of discriminant properties, and if some URI in the diseasome dataset has a value w.r.t. diseasome:name that is similar to some URI in the sider dataset w.r.t. sider:siderEffectName, these two URIs can be considered as coreferent. In this track, the training process converges at two iterations, respectively.

Table 1. Property discriminability on the DI track

	Property in dataset1	Property in dataset2
diseasome vs. sider	rdfs:label	sider:sideEffectName
	diseasome:name	sider:siderEffectName
	rdfs:label	rdfs:label
	diseasome:name	rdfs:label
dailymed vs. sider	dailymed:genericMedicine	sider:drugName
	dailymed:name	sider:drugName
	dailymed:genericMedicine	rdfs:label
	dailymed:name	rdfs:label
drugbank vs. sider	drugbank:genericName	sider:drugName
	rdfs:label	sider:drugName
	drugbank:genericName	rdfs:label
	rdfs:label	rdfs:label
	drugbank:synonym	sider:drugName
	drugbank:synonym	rdfs:label
	drugbank:pubchemCompoundId	sider:siderDrugId
	drugbank:brandName	sider:drugName

With these discriminant property pairs, ObjectCoref finds a number of coreferent URIs for each pair of datasets. As shown in Table 2, the precision and recall is moderate. Without considering the type of each object, the precision is not very good, so further inference-based debugging on coreferent URIs is needed for future work.

Table 2. Performance of ObjectCoref on the DI track

	Found	Existing	Precision	Recall	F-measure
diseasome vs. sider	190	238	0.837	0.668	0.743
dailymed vs. sider	2903	1592	0.548	0.999	0.708
drugbank vs. sider	933	283	0.302	0.996	0.464

2.2 PR

In this track, ObjectCoref uses the same self-training process to recognize coreferent URIs for each pair of datasets, two of which are related to persons and the other is about restaurants. The discriminant property pairs are listed in Table 3. Based on these discriminant properties, ObjectCoref finds a set of coreferent URIs, where the precision and recall are pretty good (see Table 4). In particular, the good recall reflects that our learning approach identifies the key properties for resolving object coreference in this track. But we also notice that some combination of properties may be also helpful. For example, first_name + last_name can be used for identifying same people.

Table 3. Property discriminability on the PR track

	Property in dataset1	Property in dataset2
person1	person11:has_address	person12:has_address
	person11:phone_number	person12:phone_number
	person11:soc_sec_id	person12:soc_sec_id
person2	person21:has_address	person22:has_address
	person21:phone_number	person22:phone_number
	person21:soc_sec_id	person22:soc_sec_id
restaurants	restaurant1:has_address	restaurant2:has_address
	restaurant1:name	restaurant2:name

Table 4. Performance of ObjectCoref on the PR track

	Found	Existing	Precision	Recall	F-measure
person1	499	500	1.000	0.998	0.999
person2	362	400	1.000	0.900	0.947
restaurants	91	112	0.989	0.804	0.887

2.3 Benchmark & conference

We use Falcon-AO (2010) to participate in the benchmark and conference tracks. The average precision and recall are depicted in Table 5. As compared to OAEI 2007, the benchmark track adds some new cases. Falcon-AO failed in several cases due to the Jena parsing errors. For the detailed results, please see Appendix.

Table 5. Performance of Falcon-AO (2010) on the benchmark and conference tracks

	Precision	Recall
Benchmark	0.76	0.64
Conference	0.60	0.60

3 General comments

In this section, we will firstly discuss several possible ways to improve ObjectCoref, and then give comments on the OAEI 2010 test cases.

3.1 Discussions on the way to improve the proposed system

The preliminary results of ObjectCoref demonstrate that using property discriminability is feasible to find coreferent URIs on the Semantic Web. However, we also see several shortcomings of the proposed approach, which will be considered in the next version.

1. *How to divide objects into different domains?* For the tasks in this year's OAEI, we may not see the importance of recognizing domains, but on the whole Semantic Web, different domains may have different discriminant properties, and a single property pair may have different discriminability in different domains. So, a uniform measurement is ineffective.

2. *How to avoid error accumulation?* In self-training, an important issue is to prevent error accumulation, since a wrong labeled example would lead to misclassification in further propagation. In our evaluation, because the training process converges in a few iterations, so this situation is not so significant. But in real world, it is imperative to consider that.

3. *How to find discriminant property combinations?* A single property may be not good enough for resolving object coreference, while the combination of several properties would be more discriminant. However, we need to avoid overfitting. So, we plan to mine frequent patterns in the RDF data for describing objects and refine these frequent patterns to form property combinations.

3.2 Comments on the OAEI 2010 test cases

The proposed matching tasks cover a large portion of real world domains, and the discrepancies between them are significant. Doing experiments on these tasks are helpful to improve algorithms and systems. In order to enhance applicability, we list some problems in our experiment procedure, which might aid organizers to improve in the future.

1. In the DI track, the organizers provide 4 downloadable datasets for the biomedical domain, however, the interlinking track also involves a number of others, e.g., linkedct, lifescience, bio2rdf. The datasets are not only very large, but also difficult to find the latest versions, most of which are even not allowed to download. Furthermore, using SPARQL endpoints in the experiment is very time-consuming, especially for such a large scale. So, we would expect that all the datasets can be (perhaps temporarily) offline in the next year.

2. Falcon-AO (2010) uses Jena 2.6.3 as the RDF parser. In the benchmark track, some ontologies may have problems and cause the Jena exception “Unqualified typed nodes are not allowed. Type treated as a relative URI”. So, we would expect the organizers to fix this in the next year.

4 Conclusion

Object coreference resolution is an important way for establishing interoperability among (Semantic) Web applications that use heterogenous data. We implement an online system for resolving object coreference called ObjectCoref, which follows a self-training framework focusing on learning property discriminability. From the experiments in this year's DI and PR tracks, we find some positive and negative experience for improving our system. In the near future, we look forward to making a stable progress towards building a comprehensive object coreference resolution system for the Semantic Web.

Acknowledgements

This work is in part supported by the NSFC under Grant 61003018 and 60773106. We would like to thank Ming Li for his valuable comments on self-training.

References

1. Berners-Lee, T., Fielding, R., Masinter, L.: Uniform Resource Identifier (URI): Generic Syntax. RFC 2396 http://www.ietf.org/rfc/rfc3986.txt
2. Bleiholder, J., Naumann, F.: Data Fusion. ACM Computing Surveys 41(1), 1–41 (2008)
3. Cheng, G., Qu, Y.: Searching Linked Objects with Falcons: Approach, Implementation and Evaluation. International Journal on Semantic Web and Information Systems 5(3): 49–70 (2009)
4. Hu, W., Qu, Y.: Falcon-AO: A Practical Ontology Matching System. Journal of Web Semantics 6(3), 237–239 (2008)
5. Jacobs, I., Walsh, N.: Architecture of the World Wide Web, Volume One. W3C Recommendation 15 December 2004. `http://www.w3.org/TR/webarch/`
6. Zhou, Z., Li, M.: Semi-supervised learning by disagreement. Knowledge and Information Systems 24(3), 415–439 (2009)

Appendix: Complete results

In this appendix, we will show the complete results of Falcon-AO (2010) on the *benchmark* and *conference* tracks. Tests were carried out on two Intel Xeon Quad 2.40GHz CPUs, 8GB memory with Redhat Linux Enterprize Server 5.4 (x64), Java 6 compiler and MySQL 5.0.

Matrix of Results

In the following tables, the results are shown by precision (Prec.) and recall (Rec.).

Bench	Description	Prec.	Rec.
101	Reference	1.00	1.00
102	Irrelevant	NaN	NaN
103	Lang. generalization	1.00	1.00
104	Lang. Restriction	1.00	1.00
201	No names	0.97	0.97
201-2		0.98	0.98
201-4		1.00	1.00
201-6		0.92	0.92
201-8		0.98	0.98
202	No names, comments	Jena error	
202-2		0.70	0.70
202-4		0.70	0.70
202-6		0.72	0.72
202-8		0.72	0.72
203	Misspelling	1.00	1.00
204	Naming conventions	0.96	0.96
205	Synonyms	0.97	0.97
206	Translation	0.94	0.94
207		0.96	0.96
208		0.98	0.98
209		0.65	0.65
210		0.68	0.68
221	No specialization	1.00	1.00
222	Flattened hierarchy	1.00	1.00
223	Expanded hierarchy	1.00	1.00
224	No instances	1.00	0.99
225	No restrictions	1.00	1.00
228	No properties	1.00	1.00
230	Flattened classes	0.94	1.00
231	Expanded classes	1.00	1.00
232		1.00	0.99
233		1.00	1.00
236		1.00	1.00
237		1.00	0.99
238		1.00	0.99
239		1.00	1.00
240		1.00	1.00
241		1.00	1.00
246		1.00	1.00
247		1.00	1.00
248		Jena error	
248-2		0.69	0.68
248-4		0.71	0.69
248-6		0.73	0.71
248-8		0.69	0.68
249		Jena error	
249-2		0.70	0.70
249-4		0.71	0.71
249-6		0.73	0.73
249-8		0.73	0.73
250		Jena error	
250-2		1.00	0.79
250-4		1.00	0.61
250-6		0.93	0.42
250-8		0.88	0.21

Bench	Description	Prec.	Rec.
251		Jena error	
251-2		0.99	0.78
252-4		0.53	0.53
252-6		0.55	0.55
252-8		0.55	0.55
253		Jena error	
253-2		0.71	0.69
253-4		0.69	0.68
253-6		0.67	0.66
253-8		0.69	0.68
254		Jena error	
254-2		1.00	0.79
254-4		1.00	0.61
254-6		0.93	0.42
254-8		0.88	0.21
257		Jena error	
257-2		1.00	0.79
257-4		1.00	0.61
257-6		0.93	0.42
257-8		0.88	0.21
258		Jena error	
258-2		0.99	0.78
258-4		1.00	0.59
258-6		0.97	0.40
258-8		0.95	0.22
259		Jena error	
259-2		0.55	0.55
259-4		0.51	0.51
259-6		0.55	0.55
259-8		0.54	0.54
260		Jena error	
260-2		0.96	0.79
260-4		1.00	0.62
260-6		0.92	0.41
260-8		0.88	0.24
261		Jena error	
261-2		1.00	0.79
261-4		1.00	0.79
261-6		1.00	0.79
261-8		1.00	0.79
262		Jena error	
262-2		1.00	0.79
262-4		1.00	0.61
262-6		0.93	0.42
262-8		0.88	0.21
265		Jena error	
266		Jena error	
301	BibTeX/MIT	0.91	0.68
302	BibTeX/UMBC	0.87	0.56
303	BibTeX/Karlsruhe	0.73	0.73
304	BibTeX/INRIA	0.95	0.92

Conference	Prec.	Rec.
cmt-conference	0.50	0.56
cmt-confof	0.55	0.38
cmt-edas	0.69	0.69
cmt-ekaw	0.55	0.55
cmt-iasted	0.50	1.00
cmt-sigkdd	0.77	0.83
conference-confof	0.53	0.60
conference-edas	0.48	0.59
conference-ekaw	0.50	0.48
conference-iasted	0.63	0.36
conference-sigkdd	0.71	0.67
confof-edas	0.45	0.53
confof-ekaw	0.62	0.65
confof-iasted	0.36	0.44
confof-sigkdd	0.80	0.57
edas-ekaw	0.65	0.57
edas-iasted	0.64	0.37
edas-sigkdd	0.88	0.47
ekaw-iasted	0.54	0.70
ekaw-sigkdd	0.78	0.64
iasted-sigkdd	0.59	0.87

An integrated matching system: GeRoMeSuite and SMB – Results for OAEI 2010

Christoph Quix[1], Avigdor Gal[2], Tomer Sagi[2], David Kensche[1]

[1]RWTH Aachen University, Germany
`http://www.dbis.rwth-aachen.de`

[2]Technion – Israel Institute of Technology
`http://www.technion.ac.il/`

Abstract. We present the results of an integrated matching system which is the result of a cooperation project between the Israel Institute of Technology (Technion) and the RWTH Aachen University in Germany. We have integrated the GeRoMeSuite system (from RWTH Aachen) and SMB (from Technion). Both tools aim at matching schemas; while GeRoMeSuite offers a variety of matchers, SMB provides the information on how to combine matchers and how to enhance match results. Thus, an integration of the tools is beneficial for both systems.

1 Presentation of the system

1.1 GeRoMeSuite

As a framework for model management, *GeRoMeSuite* [3] provides an environment to simplify the implementation of model management operators. *GeRoMeSuite* is based on the generic role based metamodel *GeRoMe* [2], 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 [6], Merge [5], and Compose [4].

The matching component of *GeRoMeSuite* has been described in more detail in [6], 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 [3].

1.2 SMB

The Schema Matching Boosting (SMB) Service is a toolkit for enhancing the performance of schema matchers. SMB operates in 3 modes: Enhance, Learn, and Recommend. In the *enhance* mode, SMB recieves a raw correspondence matrix (with similarity values for attribute correspondence in the range of [0,1]) and performs an analysis of the results per row and column. Subsequently, SMB uses contrasting and weakening algorithms to boost results of “promising” rows and columns and weaken results of “non-promising” rows and columns respectively. Contrasting is perfromed using a modified version of the Weber contrast function. Weakening is inversly proportional to the row and column average.

The *learn* mode is used to perform off-line training of SMB on the perfromance behavior of matchers w.r.t. various matching tasks which are classified to classes according to their a-priory features such as schema size. Training is performed using the SMB algorithm, as introduced in [1]. The *recommend* classifies in run-time a given matching task, providing the reccomended ensemble weights for the matching systems various components. The *Learn* and *recommend* modes are a re-implementation of the system presented in [1] in which run-time complexity has been reduced from $O(n^{'})$ to $O(n^2)$ and generic interfaces have been provided to allow any matching system to use SMB by command-line invocation.

1.3 State, purpose, general statement

GeRoMeSuite is a generic system which can match ontologies as well as schemas in other modeling languages such as XML Schema or SQL. Therefore, it is well suited for matching tasks across heterogeneous modeling languages, such as matching XML Schema with OWL. We discussed in [6] 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.

SMB is also a modeling language independent ‘meta’ matching system which mainly works on the similarity matrices produced by GeRoMeSuite. It improves the *clarity* of the similarity values by improving ‘good’ values and descreasing ‘bad’ values. This should increase the precision of the match result.

1.4 Specific techniques used

Besides the integration of GeRoMeSuite and SMB, we focused this year on adding validation methods to the system to improve the precision of the match result. A component for adding disjointness relationships in an ontology has been added to the matching framework. The component uses machine learning techniques to identify disjoint concepts with one ontology. The disjointness relationships can then be used in the validation of schema matches using logical reasoning.

Furthermore, we developed a component which can use a background ontology to find additional matches in the ontology. The system is able to find an appropriate background ontology on the web automatically, using Google and Swoogle. Due to the set up of the OAEI campaign, we did not use this component for OAEI.

1.5 Adaptations made for the evaluation

We evaluated several match configurations which is easily possible due to the adaptable and extensible matching framework of GeRoMeSuite. As only one configuration can be used for all matching tasks, we had to find a good compromise between performance in terms of precision and recall, time performance for larger ontologies (e.g., anatomy), and selection of appropriate matchers which work well on all tracks. For example, we also tested configurations which had an f-measure that was about 5% higher than the configuration which we eventually used, but these configurations did not work well on all tracks. The identification of good match configurations is a topic for future research.

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 matching only elements of the same type, e.g., classes with classes only and properties with properties only.

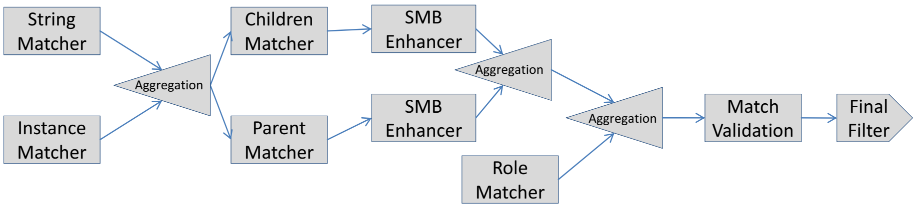


Fig. 1. Matching Strategy for OAEI 2010

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 of *GeRoMeSuite*. The command line interface can work in a batch mode in which several matching tasks and configurations can be processed and compared. The existence of this tool enabled also an easy integration with the OAEI web service interface.

1.6 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/oaei2010/`

The page provides also links to the configuration files used for the evaluation.

1.7 Link to the set of provided alignments (in align format)

The results for the OAEI campaign 2010 are available at `http://www.dbis.rwth-aachen.de/gerome/oaei2010/`

2 Results

2.1 Benchmark

The following table shows the average results for precision and recall in the benchmark track.

Task	Precision	Recall
1xx	1,00	1,00
2xx (xx<48)	0,96	0,88
2xx (xx>47)	0,89	0,51
3xx	0,79	0,38

A first check, whether a match configuration is suitable at all are the 1xx ontologies. A configuration should produce the perfect result for these tracks, which is the case for the configuration, we have finally chosen.

For the simpler tasks in the 2xx data set (201-247), our system was able to achieve a very good result with an f-measure of more than 0.9.

For some of the really difficult tasks (248-266), our system was not able to find any correspondence as there is hardly any information that can be used (e.g., task 265 with no labels, no comments, no hierarchy, etc.).

The results for the tasks 3xx was in general good (f-measure of about 0.6 for 301, 302, and 304). However, ontology 303 is difficult for our generic system as the namespaces are not defined in a standard way. Therefore, we could only find a few correspondences.

2.2 Conference

The ontologies in the conference track are rather small and the matching tasks are more difficult as the ontologies have been designed by humans using different terminologies and having different goals in mind. As this is a more realistic case than the benchmark track, we have chosen a configuration which produces good results for the conference track. Using validation rules to check the logical consistency of the identified correspondences and a final filter step which generates only 1:1 correspondences was beneficial for the quality of the result.

At the current point, we can only report the results with respect to the reference alignments which are available. For these tasks, we achieve an average f-measure of about 0.45.

2.3 Anatomy

We participated in this task in the sub-tracks 1 to 3. Probably because of our validation and filtering methods, we achieved a high precision but low recall in task 1. Therefore, we used the result of task 1 also for task 2. In task 3, we achieved a high recall with respect to the *partial* reference alignment. We have to wait for the results with respect to the full alignment to make a final statement about the quality for this subtask.

2.4 Directory

We participate only in the single task modality of the directory track. The size of the input ontologies is similar to the anatomy track, so the same problems of scalability have to be faced here. We submitted an alignment with about 700 correspondences. Due to a missing reference alignment for the single task modality, we could not evaluate the quality of this result.

The main reason for not participating in the small task modality is that the small ontologies do not contain enough information to do a reasonable matching. Furthermore, we think that many of the given reference alignments are not correct.

3 Comments

We participate this time the third time in OAEI and see again some 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. We appreciate especially the automatic evaluation system, although we also had to put some additional effort to get the interface and our web service working.

However, some reference alignments, especially in the directory track, should be reconsidered as they do not seem to be right. Furthermore, an oriented track as in OAEI 2009 would be useful to evaluate semantic matching techniques.

We are currently working on a system to generate a matching benchmark which comes closer to the challenges of real ontologies. We would be happy if we could contribute the results to OAEI 2011.

4 Conclusion

As our tool is neither specialized on ontologies nor limited to the matching task, we did not expect to deliver the best results. However, we are very satisfied with the overall results, as we can compete with the special purpose ontology alignment tools.

We will continue to work on the improvement of our matching system and on the integration of GeRoMeSuite and SMB. We will especially focus on the problem of identifying good match configurations automatically. 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. A. Gal, T. Sagi. Tuning the Ensemble Selection Process of Schema Matchers. *Information Systems*, **35**(8):845–859, 2010.
2. 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.
3. 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.
4. D. Kensche, C. Quix, Y. Li, M. Jarke. Generic Schema Mappings. *Proc. 26th Intl. Conf. on Conceptual Modeling (ER'07)*, pp. 132–148. 2007.

5. 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*, vol. 4495, pp. 127–141. Springer-Verlag, 2007.
6. 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.

LN2R – a knowledge based reference reconciliation system: OAEI 2010 Results

Fatiha Saïs[1], Nobal Niraula[2], Nathalie Pernelle[1], Marie-Christine Rousset[3]

[1] LRI (Paris-Sud 11 University & CNRS) / INRIA Saclay Parc-Club Orsay Univ.
Bat. G ,4, rue Jacques Monod F-91893 Orsay Cedex, France
{fatiha.sais, nathalie.pernelle}@lri.fr
[2] University of Memphis, TN, USA
nb.niraula@gmail.com
[3] LIG - Laboratoire dInformatique de Grenoble
BP 72, 38402 St MARTIN DHERES, France
Marie-Christine.Rousset@imag.fr

Abstract. This paper presents the first participation of LN2R system in IM@OAEI2010, the Instance Matching track of Ontology Alignment Evaluation Initiative 2010 Campaign. In particular, we participated in OWL data track by performing LN2R system on Person-Restaurant data set. We obtained very good results on person data sets and reasonable results on restaurant data set.

1 Presentation of the system

To design a semantic information integration system, we are faced to two reconciliation problems. First, the schema (or ontology) reconciliation which consists in finding mappings between elements (concepts or relations) of two schemas or two ontologies (see [1, 2] for surveys). The second problem concerns data reconciliation (named reference reconciliation) which consists in comparing data descriptions and deciding whether different descriptions refer to the same real world entity (e.g. the same person, the same article, the same gene). The problem of reference reconciliation is very critical, since it impacts data quality and data consistency [3].

In LN2R system, we address only the problem of reference reconciliation. There are several kinds of reference reconciliation approaches: knowledge-based, similarity-based, probabilistic, supervised, etc.[4]. In this paper we focus our study on reference reconciliation approaches that are *informed* and *global*. *Informed* approaches are those which exploit knowledge that is declared in the ontology to reconcile data. Reference reconciliation approaches are said *global* when they exploits the dependencies possibly existing between reference reconciliations [5, 6]. Such approaches use attribute values describing the data but also references that are related to the considered data [5, 6]. For example, the reconciliation between two scientists can entail the reconciliation between their two affiliated universities. Such dependencies result from the semantics of the domain of interest.

1.1 State, purpose, general statement

The reference reconciliation system (LN2R) that we have tested in IM@OAEI2010 campaign is knowledge-based, unsupervised and based on two methods, a logical one called L2R and a numerical one called N2R. The Logical method for Reference Reconciliation (L2R) is based on the translation in first order logic (Horn rules) of some of the schema semantics. In order to complement the partial results of L2R, we have designed a Numerical method for Reference Reconciliation (N2R). It exploits the L2R results and allows computing similarity scores for each pair of references.

Reference reconciliation problem. Let $S1$ and $S2$ be two data sources which conform to the same OWL ontology. Let $I1$ and $I2$ be the two reference sets that correspond respectively to the data of $S1$ and $S2$. The problem consists in deciding whether references are reconciled or not reconciled. Let $Reconcile$ be a binary predicate. $Reconcile(X, Y)$ means that the two references denoted by X and Y refer to the same world entity. The reference reconciliation problem considered in L2R consists in extracting from the set $I1 \times I2$ of reference pairs two subsets REC and $NREC$ such that: $REC = \{(i, i), Reconcile(i, i)\} \quad and \quad NREC = \{(i, i), \neg Reconcile(i, i)\}$

The reference reconciliation problem considered in N2R consists in, given a similarity function $Simr : I1 \times I2 \rightarrow 0..1$, and a threshold Trec (a real value in 0..1 given by an expert, fixed experimentally or learned on a labeled data sample), computing the following set:
$REC_{N2R} = \{(i, i') \in (I_1 \times I_2) \backslash (REC \cup NREC), s.t. Sim_r(i, i') > T_{rec}\}$

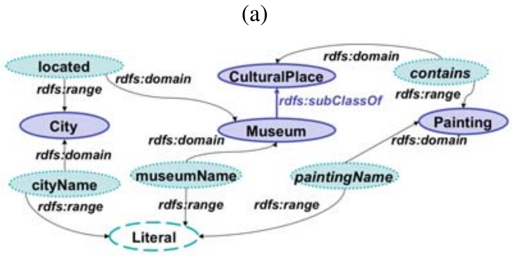
1.2 Specific techniques used

In the following, we will present some details on the knowledge-based reference reconciliation system (LN2R). First, we will show through an example the ontology and the kind of knowledge that we use. Second, we give a brief presentation of the two methods L2R and N2R of reference reconciliation.

The ontology and its constraints The considered OWL ontology consists of a set of classes (unary relations) organized in a taxonomy and a set of typed properties (binary relations). These properties can also be organized in a taxonomy of properties. Two kinds of properties can be distinguished in OWL: the so-called relations (in OWL abstractProperty), the domain and the range of which are classes and the so-called attributes (in OWL objectProperty), the domain of which is a class and the range of which is a set of basic values (e.g. Integer, Date, Literal).

We allow the declaration of constraints expressed in OWL-DL or in SWRL in order to enrich the domain ontology. The constraints that we consider are of the following types:

- Constraints of disjunction between classes: DISJOINT(C,D) is used to declare that the two classes C and D are disjoint.
- Constraints of functionality of properties: PF(P) is used to declare that the property P (relation or attribute) is a functional property.



(b)

Source S1:
MuseumName(S1_m1,”Le Louvre”); Contains(S1_m1,S1_p1); Located(S1_m1,S1_c1); CityName(S1_c1,”Paris”); PaintingName(S1_p1, ”La Joconde”);

Source S2:
MuseumName(S2_m1,”muse du Louvre”); Located(S2_m1,S2_c1); Contains(S2_m1,S2_p1); Contains(S2_m1,S2_p2);CityName(S2_c1,”Ville de paris”); PaintingName(S2_p1, ”Abricotiers en fleurs”); PaintingName(S2_p2,”Joconde”);

Fig. 1. (a) an extract of cultural place ontology, (b) an extract of RDF data

- Constraints of inverse functionality of properties: PFI(P) is used to declare that the property P (relation or attribute) is an inverse functional property. These constraints can be generalized to a set $\{P_1, \ldots, P_n\}$ of relations or attributes to state a combined constraint of inverse functionality that we will denote PFI($P_1, \ldots, P_n$). For example, $PFI(located, name)$ expresses that one address and one name cannot be associated to several cultural places (i.e. both are needed to identify a cultural place).

Data description and their constraints A piece of has a reference, which has the form of a URI (e.g. `http://www.louvre.fr,NS-S1/painting243`), and a description, which is a set of RDF facts involving its reference. An RDF fact can be: either (i) a class-fact $C(i)$, where C is a class and i is a reference, (ii) a relation-fact $R(i1, i2)$, where R is a relation and $i1$ and $i2$ are references, or (iii) an attribute-fact $A(i, v)$, where A is an attribute, i a reference and v a basic value (e.g. integer, string, date).

The data description that we consider is composed of the RDF facts coming from the data sources enriched by applying the OWL entailment rules. We consider that the descriptions of data coming from different sources conform to the same OWL ontology (possibly after schema reconciliation). Figure 1 (b), provides examples of data coming from two RDF data sources S1 and S2, which conform to a same ontology describing the cultural application previously mentioned.

L2R: a Logical method for Reference Reconciliation L2R [7] is based on the inference of facts of reconciliation (Reconcile(i,j)) and of non-reconciliation ($\neg Reconcile(i', j')$) from a set of facts and a set of rules which transpose the semantics of the data sources and of the schema into logical dependencies between reference reconciliations. Facts of synonymy (SynVals(v1,v2)) and of no synonymy ($\neg SynVals(u1, u2)$) between basic values (strings, dates) are also inferred. For instance, the synonymy SynVals(“JoDS”, “Journal of Data Semantics”) may be inferred.

The L2R distinguishing features are that it is global and logic-based: every constraint declared on the data and on the OWL ontology is automatically translated into first-order logic Horn rules (rules for short) that express dependencies between reconciliations. For instance, the following rule R translates the knowledge that the two classes $Museum$ and $City$ are disjoint, $R : Museum(X) \wedge City(Y) \Rightarrow \neg Reconcile(X, Y)$

To deduce all the (non) reconciliation and (non) synonymy facts from the knowledge base, we use a logical reasoning based on the unit-resolution inference rule. The advantage of such a logical approach is that if the data are error-free and if the declared constraints are valid, then the reconciliations and non-reconciliations that are inferred are correct, thus guaranteeing a 100 % precision of the results.

N2R: a Numerical method for Reference Reconciliation N2R [5] has two main distinguishing characteristics. First, it is fully unsupervised: it does not require any training phase from manually labeled data to set up coefficients or parameters. Second, it is based on equations that model the influence between similarities. In the equations, each variable represents the (unknown) similarity between two references while the similarities between values of attributes are constants. These constants are obtained, either (i) by exploting a dictionnary of synonyms (e.g. WordNet thesaurus, the dictionnary of synonyms generated by L2R [7]); or (ii) by using standard similarity measures on strings or on sets of strings. Furthermore, ontology and data knowledge (disjunctions and UNA) is exploited by N2R in a filtering step to reduce the number of reference pairs that are considered in the equation system. The functions modeling the influence between similarities are a combination of maximum and average functions in order to take into account the constraints of functionality and inverse functionality declared in the OWL ontology in an appropriate way.

The equations modeling the dependencies between similarities. For each pair of references, its similarity score is modeled by a variable x_i and the way it depends on other similarity scores is modeled by an equation: $x_i = f_i(X)$, where $i \in [1..n]$ and n is the number of reference pairs for which we apply N2R, and $X = (x_1, x_2, \ldots, x_n)$. Each equation $x_i = f_i(X)$ is of the form: $f_i(X) = max(f_{i-df}(X), f_{i-ndf}(X))$

The function $f_{i-df}(X)$ is the maximum of the similarity scores of the value pairs and the reference pairs of attributes and relations with which the i-th reference pair is functionally dependent. The maximum function allows propagating the similarity scores of the values and the references having a strong impact. The function $f_{i-ndf}(X)$ is defined by a weighted average of the similarity scores of the values pairs (and sets) and the reference pairs (and sets) of attributes and relations with which the i-th reference pair is not functionally dependent. See [5] for the detailed definition of $f_{i-df}(X)$ and $f_{i-ndf}(X)$.

Iterative algorithm for reference pairs similarity computation. Solving this equation system is done by an iterative method inspired from the Jacobi method [8], which is fast converging on linear equation systems. To compute the similarity scores, we have implemented an iterative resolution method. At each iteration, the method computes the variable values by using those computed in the precedent iteration. Starting from an initial vector $X^0 = (x_1^0, x_2^0, ..., x_n^0)$, the value of the vector X at the k-th iteration is obtained by the expression: $X^k = F(X^{k-1})$. At each iteration k we compute the value of each x_i^k: $x_i^k = f_i(x_1^{k-1}, x_2^{k-1}, ...x_n^{k-1})$ until a fix-point with precision ϵ is reached. The fix-point is reached when: $\forall i, |x_i^k - x_i^{k-1}| <= \epsilon$.

The similarity computation is illustrated by the equation system (see Table 1) obtained from the data descriptions shown in the example 1.

- x_1 = Sim_r(S1_m1, S2_m1) ; Sim_v(“Le louvre”, “Musee du louvre”) = 0.68
- x_2 = Sim_r(S1_p1, S2_p1) ; Sim_v(“La Joconde”, “Abricotiers en fleurs”) = 0.1
- x_3 = Sim_r(S1_p1, S2_p2) ; Sim_v(“La Joconde”, “Joconde”) = 0.9
- x_4 = Sim_r(S1_c1, S2_c1) ; Sim_v(“Paris”, “Ville de Paris”) = 0.42

The weights are computed in function of the number of common attributes and common relations of the reference pairs. The weights used in the value computation of the variables x_1, x_2, x_3 and x_4 are respectively: $\lambda_{11} = 1/4$, $\lambda_{21} = 1/2$, $\lambda_{31} = 1/2$ and $\lambda_{41} = 1/2$. We assume that point-fix precision ϵ is equal to 0.005.

The equation system is the one given in the example 2. The different iterations of the resulting similarity computation are provided in Table 1.

Iterations	**0**	**1**	**2**	**3**	**4**
$x_1 = \max(0.68, x_2, x_3, \frac{1}{4} * x_4)$	0	0.68	0.9	0.9	0.9
$x_2 = \max(0.1, \frac{1}{2} * x_1)$	0	0.1	0.34	0.45	0.45
$x_3 = \max(0.9, \frac{1}{2} * x_1)$	0	0.9	0.9	0.9	0.9
$x_4 = \max(0.42, x_1)$	0	0.42	0.68	0.9	0.9

Table 1. *Example of iterative similarity computation*

The solution of the equation system is $X = (0.9, 0.45, 0.9, 0.9)$. This corresponds to the similarity scores of the four reference pairs. The fix-point has been reached after four iterations. If we fix the reconciliation threshold T_{rec} at 0.80, then we obtain three reconciliation decisions: two cities, two museums and two paintings.

1.3 Adaptations made for the evaluation

In order to perform the evaluation of LN2R system on the data sets provided by IM@OAEI2010 evaluation campaign we were faced to do some choices and adaptations. As LN2R system assume that the data sets conform to the same ontology, we have performed the following steps:

1. manual alignment of the two ontologies (schemas),
2. choose a federated ontology (one among the two considered ontologies) and
3. transform the other ontology to the chosen one.

Thanks to the small size of the considered ontologies and to their structural and semantic closeness, the above steps were performed easily. For example, for the restaurant data set, the difference between the two ontologies is that the $category$ and $city$ are object properties in one ontology and data properties in the other.

In addition to this, we have transformed the LN2R output to comply with the OAEI alignment format.

1.4 Link to the system and parameters file

The LN2R system (including the parameters file) can be downloaded at:
`http://www.lri.fr/~sais/IM-OAEI10/LN2RSystem.zip`.

1.5 Link to the set of provided alignments (in align format)

The results that are obtained by LN2R in the instance matching track of OAEI 2010 campaign can be found at:
`http://www.lri.fr/~sais/IM-OAEI10/LN2RResults.zip`.

2 Results

In this section we present our comments on the results obtained from the first participation of LN2R in the Instance matching track IM@OAEI 2010. We have tested LN2R system on person and restaurant (PR) data sets.

To evaluate our system we have compared its results on the different data sets with the provided gold-standard and we have computed the recall, the precision and the F-measure.

2.1 Person1 and Person2 data sets

In this track, participants are asked to find all correct alignments between person' instances for the two data sets $person1$ and $person2$. Each data set contains the OWL ontologies, the two RDF files to be reconciled and the reference alignments (gold-standard) file.

Since, in LN2R, the reconciliation decisions are based on a reconciliation threshold, we have performed several tests by varying the threshold value from 0.6 to 1. The best results are obtained for a threshold of 0.75 for both data sets: (i) for $person1$ data set, we have obtained the maximum F-measure of 100 % for a recall of 100 % and a precision of 100% and (ii) for the $person2$ data set we have obtained a F-measure of 93% for a recall of 88.25 % and a precision of 99.4 %.

Furthermore, as our method is global in the sense that the reconciliation decisions between instances are propagated to other pairs of instances through the relations which link them together, we have also inferred alignments between address instances. Thanks to the functionnality of $has - address$ property, we have obtained 500 alignments between address instances for $person1$ data set and 355 alignments for $person2$ data set for a threshold of 0.75.

In addition to reconciliations (positive alignments), we also infer non reconciliations between instances, thanks to the reasoning on ontology knowledge, like disjunctions and UNA. In $person1$ and $person2$ ontologies we have declared the disjunction between $person$ and $address$ classes which leads to the inference of a non reconciliation between every person instance and every address instance. These non reconciliations are very useful in a filtering step where unnecessary comparisons and similarity computation are avoided.

2.2 Restaurant data set

In this track, participants are asked to find all correct alignments between restaurant' instances of the $restaurant1$ data set. It contains the OWL ontologies, the two RDF files to be reconciled and the reference alignments (gold-standard) file.

As we have done for the person data sets, we have also performed several tests of the system by varying the threshold value from 0.6 to 1. Comparing to the gold standard, the best results are obtained for a threshold of 0.85. We have obtained a F-measure of 75.3 % for a recall of 75 % and a precision of 75.67 %.

Similarly to the person data sets, we have also inferred a set of alignments between address instances thanks to the propagation mechanism. In a filtering step, we have also inferred a set of non reconciliation (negative alignments) between restaurant and address instances.

By analyzing the results of the restaurant data set, we have noticed some mistakes in the provided reference alignments: correct alignments are missed (see example 1) and some given alignments are wrong (see example 2).
Example 1: the two instances (http://www.okkam.org/oaie/restaurant1-Restaurant16, http://www.okkam.org/oaie/restaurant2-restaurant26) should be included in the gold-standard because they refer to the same restaurant. There descriptions are as follows:
(1) ['name: patina', 'category: californian', 'phone_number:213/467-1108' , has-address[street:'5955 melrose ave.', is_in_city[name:los angeles']]]
(2) ['name: patina', has_category['name:californian'], 'phone_number:213-467-1108' , has-address['city:los angeles', street:'5955 melrose ave.']]

Example 2: the two instances (http://www.okkam.org/oaie/restaurant1-Restaurant2, http://www.okkam.org/oaie/restaurant2-restaurant2) should be removed from the gold-standard because they do not refer to the same restaurant. There descriptions are as follows:
(1) ['name: hotel bel air', 'category: californian', 'phone_number: 310/472-1211' , has-address[street:'701 stone canyon rd.', is_in_city[name:bel air']]]
(2) ['name: art's deli', has_category['name:delis'], 'phone_number:818-762-1221' , has-address['city:studio city', street:'12224 ventura blvd.']]

3 General comments

3.1 Comments on the results

The main strength of our system is its capacity to ensure a good precision in the results. In the person data set, it shows its strength over ontology knowledge reasoning and similarity measures adaptation. LN2R system is also able to minimize the number of comparisons thanks to the filtering step which leads to improvement in running time. The weak points are: the absence of knowledge on the functionality of properties impacts the performance of the system. LN2R system works on data sets which should conform to the same ontology are for which the ontology alignment is already performed.

3.2 Discussions on the way to improve the proposed system

Our system may be improved by several ways. We are studying how LN2R can be extended to take into account alignments between classes and properties. We also want to optimize the system in order to insure its scalability.

3.3 Comments on the OAEI 2010 test cases

It will be interesting to provide test cases where the alignment inference is global. It means that the alignments may concern several kinds of entities e.g. persons, addresses, books, etc. It will be useful also to have data sets which conform to the same ontology or at least give the alignments between their corresponding ontologies.

4 Conclusion

Instance matching is very important to realize the semantic Web ambitions by facilitating interoperability of ontology based applications. In this paper, we have presented the promising results of LN2R system for its first participation in the instance matching track of OAEI 2010. By this experience, we have shown LN2R strengths when the ontology knowledge is rich. In the person data sets, LN2R has obtained very good results and reasonable ones for the restaurant data sets. As future work, we will study the extension of LN2R to the general problem of matching ontologies with instances. We also plan to optimize LN2R by designing a distributed inference algorithm.

References

1. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. (2005) 146–171
2. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal **10**(4) (2001) 334–350
3. Batini, C., Scannapieco, M.: Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
4. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: A survey. IEEE Trans. on Knowl. and Data Eng. **19**(1) (2007) 1–16
5. Saïs, F., Pernelle, N., Rousset, M.C.: Combining a logical and a numerical method for data reconciliation. J. Data Semantics **12** (2009) 66–94
6. Dong, X., Halevy, A.Y., Madhavan, J.: Reference reconciliation in complex information spaces. In: SIGMOD Conference. (2005) 85–96
7. Saïs, F., Pernelle, N., Rousset, M.C.: L2r: A logical method for reference reconciliation. In: AAAI. (2007) 329–334
8. Golub, G.H., Loan, C.F.V.: Matrix computations (3rd ed.). Johns Hopkins University Press, Baltimore, MD, USA (1996)
9. Cohen, W.W., Ravikumar, P.D., Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. In: IIWeb. (2003) 73–78

MapPSO Results for OAEI 2010

Jürgen Bock[1]

FZI Forschungszentrum Informatik, Karlsruhe, Germany
`bock@fzi.de`

Abstract. This paper presents and discusses the results produced by the MapPSO system for the 2010 Ontology Alignment Evaluation Initiative (OAEI). MapPSO is an ontology alignment approach based on discrete particle swarm optimisation (DPSO). Firstly, specific characteristics of the MapPSO system and their relation to the results obtained in the OAEI are discussed. Secondly, the results for the *benchmarks* and *directory* tracks are presented and discussed.

1 Presentation of the system

With the 2008 OAEI campaign the MapPSO system (Ontology **Map**ping by **P**article **S**warm **O**ptimisation) was introduced [1] as a novel approach to tackle the ontology alignment problem by applying the technique of particle swarm optimisation (PSO).

1.1 State, purpose, general statement

The development of the MapPSO algorithm has been motivated by the following observations:

1. Ontologies are becoming numerous in number and large in size.
2. Ontologies evolve gradually.
3. Ontologies differ in key characteristics that can be exploited in order to compute alignments.

Solving the ontology alignment problem using a PSO-based approach, as done by the MapPSO system, tackles these observations as follows:

1. PSO works inherently parallel, such that large ontologies can be aligned on a parallel computation infrastructure.
2. PSO works incrementally, which allows the algorithm to start with an initial or partial configuration (*i.e.* for instance an alignment of previous ontology versions) and refine it as the ontologies evolve.
3. PSO works as a meta-heuristic, *i.e.* independently of the objective function to be optimised. In the case of ontology alignment this means that the objective function can be adjusted according the particular alignment scenario at hand.

The idea of the MapPSO approach is to provide an algorithm that fulfils the aforementioned characteristics. Particularly the focus is not to provide a universal library of similarity measures (base matchers) to form that specific objective function to be optimised, but rather to provide a scalable mechanism that can used with various objective functions depending on the alignment scenario at hand.

MapPSO is still in the status of a research prototype, where recent work has been done exploiting the parallel nature of the algorithm in a cloud-based infrastructure [2].

1.2 Specific techniques used

MapPSO treats the ontology alignment problem as an optimisation problem and solves it by applying a discrete particle swarm optimisation (DPSO) algorithm [3]. To this end, each particle in the swarm represents a valid candidate alignment, which is updated in an iterative fashion. In each iteration, knowing about the particle representing the best alignment in the swarm, other particles adjust their alignments, influenced by this best particle. A random component when adjusting an alignment makes sure that the swarm does not converge to a local optimum.

In MapPSO the quality of an alignment is determined by the average of the qualities of its correspondences, as well as by the number of correspondences in the alignment[1]. Each correspondence is evaluated by a number of base matchers, whose evaluation values are aggregated by a specified aggregator. Base matchers and aggregator can be selected via the `params.xml` configuration file. This mechanism makes MapPSO highly adjustable, since different alignment scenarios will most likely require different base matchers in order to determine similarity between entities. By following the instructions in the MapPSO documentation[2] one can easily develop base matchers and aggregators tailored to a particular alignment scenario at hand.

1.3 Adaptations made for the evaluation

Some OAEI tracks do not evaluate correspondences of all entity types. For instance in the *benchmarks* track, no instance correspondences are part of the reference alignments, while in the *instance matching* track **only** instance correspondences are part of the reference alignments. For this reason, an additional parameter was introduced for the MapPSO command-line interface that allows

---

[1] Apart from striving for correct correspondences, it is necessary to identify the correct number of correspondences, which is done in MapPSO by preferring larger alignments to smaller ones.

[2] `http://sourceforge.net/apps/mediawiki/mappso/index.php?title=Guide_for_implementing_base_matchers` and `http://sourceforge.net/apps/mediawiki/mappso/index.php?title=Guide_for_implementing_aggregation_functions` respectively.

the user to specify which correspondence types are to be included in the produced alignment.

1.4 Link to the system and parameters file

The release of MapPSO (`MapPSO.jar`) and the parameter file `params.xml` used for OAEI 2010 are located in `MapPSO.zip` at `http://sourceforge.net/projects/mappso/files/` in the folder `oaei2010`.

1.5 Link to the set of provided alignments (in align format)

The alignments of the OAEI 2010 as provided by MapPSO are located in the file `alignments.zip` at `https://sourceforge.net/projects/mappso/files/` in the folder `oaei2010`.

2 Results

The *benchmarks* track was via the (preliminary) SEALS platform[3]. To this end MapPSO has been provided as a web service[4] For the *directory* track, results were computed offline and sent to the track organiser.

2.1 benchmark

As from last year's participation it became apparent that MapPSO performs better with respect to relaxed precision and recall measures than with respect to classical measures [4, 5]. For this reason, the symmetric precision and recall measures were computed[5] in addition to the classical measures as provided by the SEALS platform. Figures 1 and 2 illustrate classical and symmetric precision, and classical and symmetric recall respectively. 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. The plots in Figures 1 and 2 were generated in a different run, than in the results obtained using the SEALS platform, thus results might not match completely.

The reason for MapPSO performing significantly better in terms of symmetric precision and recall is due to the fact that the algorithm keeps *good* correspondences not allowing them to be discarded in a later iteration. This, however, prevents entities participating in such *good* correspondences to participate in an even *better* correspondence in a later iteration. In case this *better* correspondence would be the correct one with respect to the reference alignment, the *good* one found is counted as wrong with respect to classical evaluation metrics, while its closeness is respected in the relaxed metrics.

---

[3] `http://seals.inrialpes.fr/platform/`

[4] Web service end point: `http://krake16.perimeter.fzi.de:8080/MapPSOWS`

[5] Relaxed precision and recall measures were computed using the methods provided by the Alignment API.

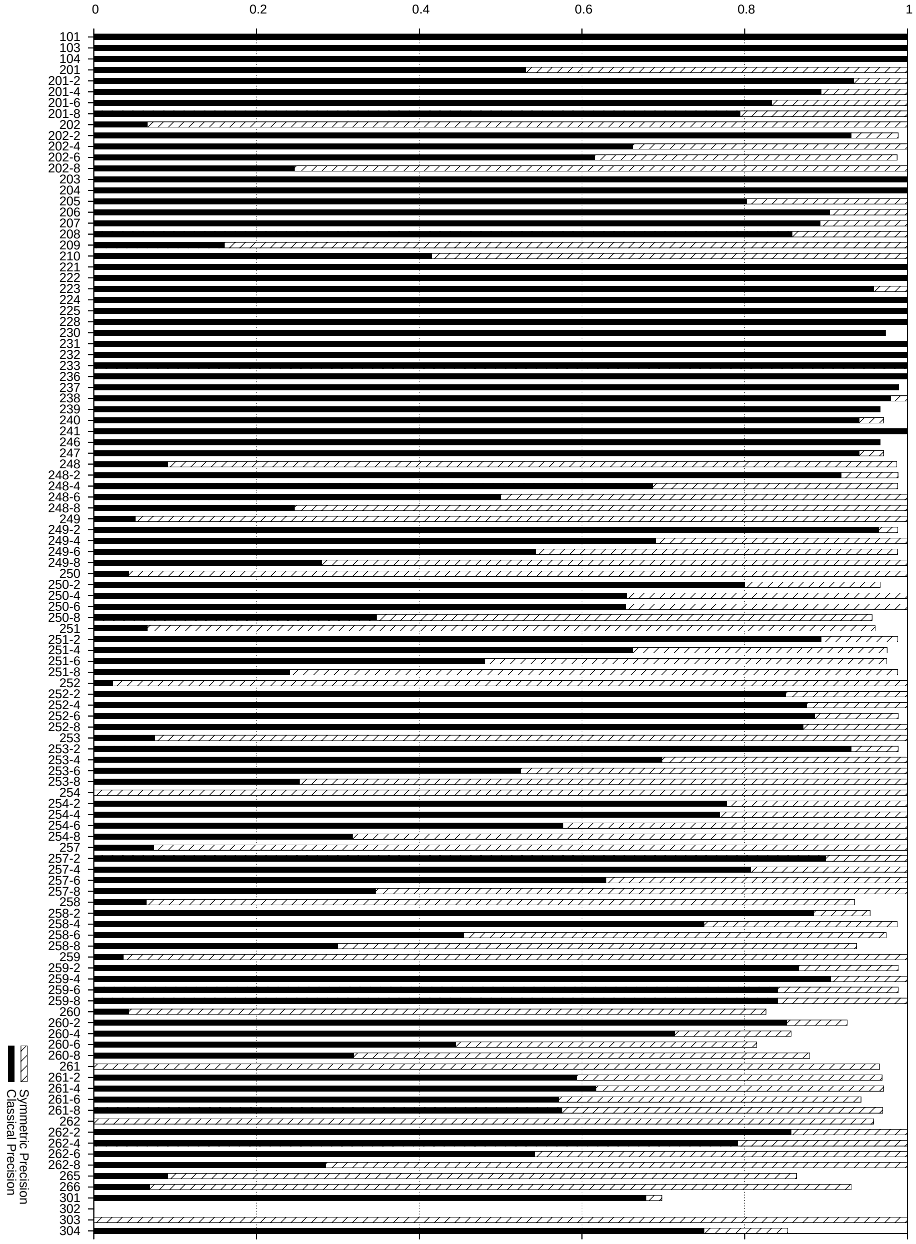


Fig. 1. Classical vs. Symmetric Precision from OAEI 2010.

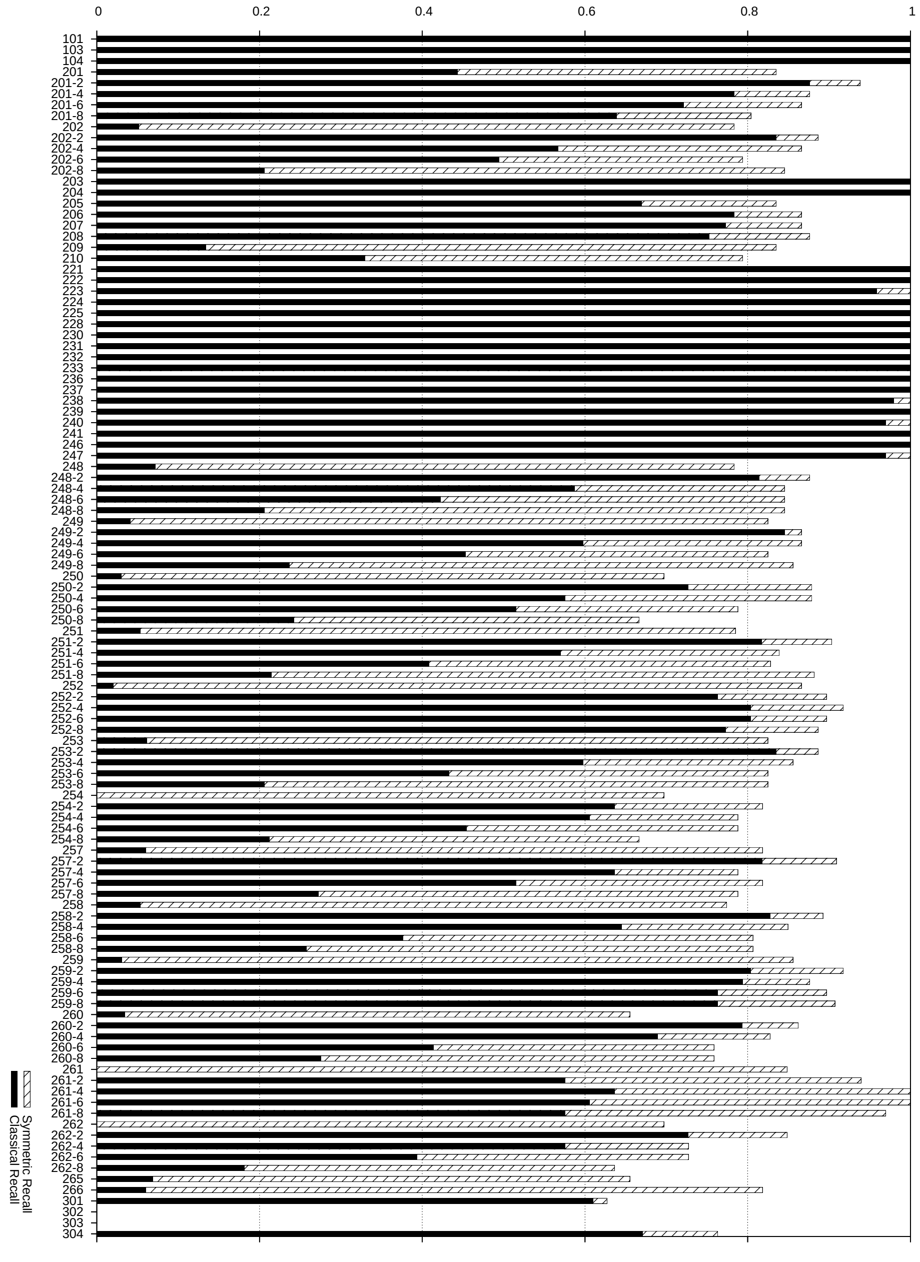


Fig. 2. Classical vs. Symmetric Recall from OAEI 2010.

2.2 directory

In the *directory* track the same set of parameters was used as in the other tracks, which includes the same set of base matchers. Due to the nature of the datasets in this track, several base matchers were not applicable or might even have contributed in a counterproductive way. For instance in the *singletask* subtrack, meaningless IDs were used as URI fragments, which could lead to a high similarity of those entities with a similar ID with respect to a particular base matcher. Since this information is known before running the matcher, deactivating this base matcher might have lead to better results. Additionally, MapPSO does not filter final results according to the confidence values gained. Thus, in its current implementation, many bad correspondences are left in the final alignment, reducing precision.

3 General comments

In the following some general statements about the OAEI procedure, modalities, and results obtained are given.

3.1 Comments on the results

Compared to the results of the 2009 *benchmarks* track, a slight decrease in the symmetric precision and recall measures can be observed. This is due to the fact that this year, MapPSO has been configured with a stronger focus on finding the correct size of an alignment. The configuration used in 2009 was rather tailored to the *benchmarks* track where the alignment is known to contain all entities. Disregarding this assumption causes the symmetric recall measure to drop, but makes the system more suitable for real-world use cases.

3.2 Discussions on the way to improve the proposed system

MapPSO is currently being worked on in order to incorporate a guided search component for two reasons. Firstly, it is expected to increase convergence speed, and secondly it is expected to improve classical precision and recall due to the reasons explained in Sect. 2.1.

3.3 Comments on the OAEI 2010 procedure

The OAEI modalities require participating systems to use the same parameter configuration for each track and each test case. According to assumption 3 stated in Sect. 1.1 different alignment scenarios will most likely require different means of determining a good alignment. Assuming that an alignment tool will not used in an out-of-the-box configuration in any real-world alignment task, makes this requirement of a single (and thus compromised) parameter configuration rather artificial.

4 Conclusion

The MapPSO system was described briefly with respect to the idea behind its DPSO-based approach. The results obtained by the MapPSO system for the OAEI 2010 tracks *benchmarks*and *directory* were presented. Several observations regarding these results were highlighted, in particular the significant difference between classical and symmetric precision and recall. Also the effect of having a single configuration throughout all OAEI tracks were discussed.

Future development of MapPSO will be targeted towards user interaction as well as alignment refinement. As for the latter, an initial (partial) alignment, such as a previous version of an alignment, can be given as a start configuration of a particle, which is then refined by running the algorithm.

Acknowledgements

The author would like to thank Michael Mutter and Carsten Dänschel for their valuable contributions and feedback.

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. Bock, J., Lenk, A., Dänschel, C.: Ontology Alignment in the Cloud. In: Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), http://ceur-ws.org, CEUR Workshop Proceedings (November 2010)
3. Bock, J., Hettenhausen, J.: Discrete Particle Swarm Optimisation for Ontology Alignment. Information Sciences **Article in Press** (2010)
4. Ehrig, M., Euzenat, J.: Relaxed Precision and Recall for Ontology Matching. In Ashpole, B., Ehrig, M., Euzenat, J., Stuckenschmidt, H., eds.: Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies. Volume 156 of CEUR Workshop Proceedings., CEUR-WS.org (October 2005) 25–32
5. Euzenat, J., Ferrara, A., Hollink, L., Isaac, A., Joslyn, C., Malaisé, V., Meilicke, C., Nikolov, A., Pane, J., Sabou, M., Scharffe, F., Shvaiko, P., Spiliopoulos, V., Stuckenschmidt, H., Šváb-Zamazal, O., Svátek, V., dos Santos, C.T., Vouros, G., Wang, S.: Results of the Ontology Alignment Evaluation Initiative 2009. In Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Noy, N., Rosenthal, A., eds.: Proceedings of the 4th International Workshop on Ontology Matching (OM-2009). Volume 551 of CEUR Workshop Proceedings., http://ceur-ws.org (October 2009)

Results of NBJLM for OAEI 2010

Song Wang[1,2], Gang Wang[1] and Xiaoguang Liu[1]

[1]College of Information Technical Science, Nankai University Nankai-Baidu Joint Lab,
Weijin Road 94, Tianjin, China
[2]Military Transportation University, The Equipment support Department, Tianjin, China
jackws66@yahoo.com

Abstract. This paper presents the results obtained by NBJLM (Nankai Baidu Joint Lab Matcher) for its first participation to OAEI 2010. The research of ontology-based similarity calculation among concepts has already been a hot issue. NBJLM is an hybrid ontology alignment method that considers both similarity of literal concept and semantic structure. Simultaneously, how to accelerate matching has been mentioned in this paper and the experimental results show the remarkable improvement of matching speed. In OAEI 2010, NBJLM submitted the result for one alignment task: anatomy.

1 Presentation of NBJLM

In recent years, Ontology matching is mainly used in ontology integration, ontology merging, and ontology reusing. Many approaches to ontology matching have been proposed over the years, references[1][5][3] make full use of information, probability and statistics theory, however, they have limited ability to distinguish semantic differences, and the similarity calculation methods are not perfect. Besides, references[2][4][7][8] have considered various factors, but they do not take into account how to avoid unnecessary calculation to shorten computing time in mapping large-scale ontologies. NBJLM is a multiple strategy dynamic ontology matching system implemented in java. It considers both the literal concept and ontology structure that includes node depth, node density and semantic distance.

1.1 State, purpose, general statement

Given two heterogeneous ontologies O1 and O2, a matching is made up of a set of correspondences between pairs of node IDs belonging to O1 and O2, respectively. NBJLM is designed to find out relations of equivalence and subsumption between entities, i.e. classes and properties, issued from two ontologies. Our approach makes use of the matching strategy that considers literal similarity measure and ontology structure similarity measure. The core contributions of NBJLM is described as followed: Firstly, it uses Hash mapping algorithm to improve efficiency of calculation. Secondly, it takes a full analysis of a number of issues to be considered in structure matching, which makes the algorithm works better, and the matching results are more accurate and efficient. As demonstrated by the experimental results, our method can greatly cut the running time, meanwhile, precise matching results can be obtained.

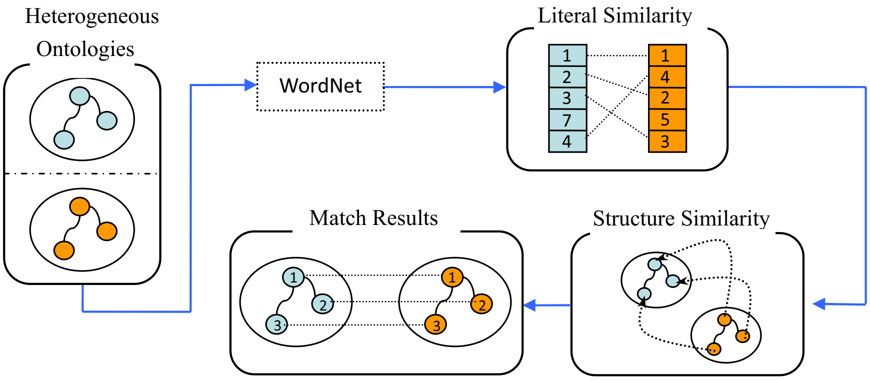


Fig. 1. Procedure of the matching of heterogeneous ontologies.

1.2 Specific techniques used for Anatomy Track

NBJLM uses a new matching strategy that considers literal similarity measure and ontology structure similarity, simultaneously. We obtain the following formula:

$$Sim(ID1, ID2) = \theta \times Sim\_literal(ID1, ID2) + (1 - \theta) \times Sim\_struct(ID1, ID2)$$

where $Sim\_literal(ID1, ID2)$ is the literal concept similarity measure, $Sim\_struct(ID1, ID2)$ is the structural similarity measure, and θ $(0 < \theta < 1)$ is paramater to control how much literal and ontology structure contribute to the ontologies matching respectively. Firstly, the measure of literal similarity is a preliminary matching. It takes account of polysemy and synonym of a word, by transforming the word into a semantic collection using WordNet. Then we can get the preliminary matching results that is semantic mapping rather than spelling mapping of words. Secondly, based on the literal matching results, the measure of ontology structure similarity is calculated through the relation between hypernym and hyponym of a word, considering distance of edges, and depth and density of node in the hierarchy of ontology. With the final combination of the two values, and with adjustment of the parameter, we could obtain more reasonable matching results. The procedure is shown in Fig. 1.

An optimized algorithm for concept sets retrieving If look up a word in WordNet, we can get one or more Synsets (defined by WordNet). For one thing each Synset is a concept set of the words which have the same meaning. For another a word may have several meanings, therefore, each Synset can be used to express one concept of the word. The concept of a node ID in the hierarchy of ontology may be described by several phrases, which are composed of words. That means the concept of the node ID could be described by several Synsets. If we deal with all the Synsets in matching, redundant computation will be inevitable. Therefore, this paper proposes a strategy that obtain the set of Synsets, which are the most similar to the concept of the phrase while

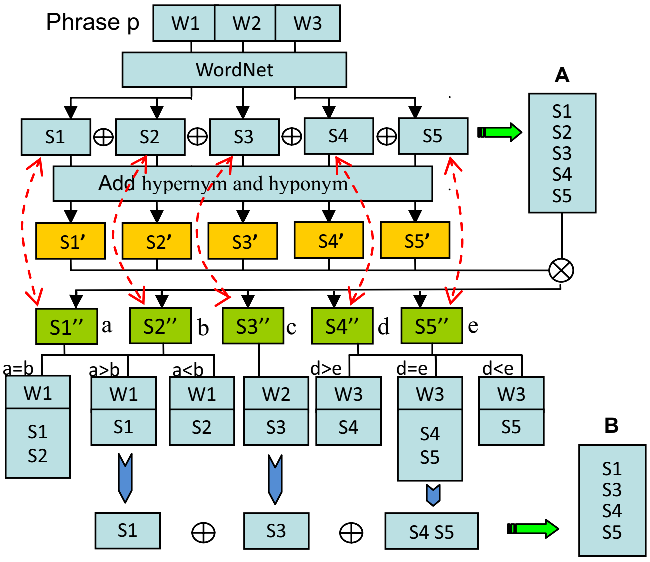


Fig. 2. Get the optimal Synsets to describe the concept of a phrase

include the least Synsets, to describe the concept of a node ID with the help of WordNet. So the unnecessary computation work could be reduced. Fig. 2 describes a simple example that how to tackle a phrase to get the optimal Synsets:

1) Obtain the Synsets of all the words $w1$, $w2$ and $w3$ from phrase p got from a node of $O1$(or $O2$) by WordNet, respectively, $(S1, S2)$, $(S3)$ and $(S4, S5)$.
2) Get the union set of all the Synsets, $A(S1, S2, S3, S4, S5)$, which denotes the concept of phrase p and includes the most Synsets.
3) Add the semantic environment (hypernym and hyponym of the Synset) to Synset. Then we get $S1'$, $S2'$, $S3'$, $S4'$ and $S5'$.
4) Intersect $A(S1, S2, S3, S4, S5)$ with $S1'$, $S2'$, $S3'$, $S4'$ and $S5'$, separately, resulting in $S1''$, $S2''$, $S3''$, $S4''$ and $S5''$. And the numbers of elements of the intersections are a, b, c, d and e. Meanwhile, establish correspondences between $S1$ and $S1''$, $S2$ and $S2''$, $S3$ and $S3''$, $S4$ and $S4''$, $S5$ and $S5''$. The purpose of doing intersections is to find correlation between semantic environment of a Synset and the concept of phrase $\boldsymbol{p}$. The larger number of the intersection's elements is, the more similar relationship between them is.
5) Compare the numbers of intersections' elements mentioned at step 4, which are generated from the same word. And select the Synset of each word, associated with the result of intersection which has the larger number of elements. For example, on the assumption that $a > b$, $d = e$ (c has no comparable object), the Synsets of $w1$, $w2$ and $w3$ are $(S1)$, $(S3)$ and $(S4, S5)$, respectively.
6) Get the union set of $(S1)$, $(S3)$ and $(S4, S5)$, $B(S1, S3, S4, S5)$, which denotes the concept of phrase $\boldsymbol{p}$.

7) It can be found that Synset $S2$ existing in A but not in B is uncorrelated to the concept of phrase p. Therefore, the redundancy can be filtered out by our optimized algorithm. Besides, as increasing in the number of words of phrase, the optimization of the algorithm could be more obvious. Since the matching of nodes in the ontologies is based on the matching of Synsets, the reduction of Synsets, which denote the concepts of nodes in the ontology, will inevitably lead to the reduction of irrelevant semantic mappings and greatly reduce the amount of calculation.

Method of calculation of structural similarity The calculation of structural similarity involves semantic distance with weight, information content, depth and density of node. In order to tackle two ontologies conveniently, we add a virtual common root node which connects two ontologies. So the model could be changed from two independent ontologies to a large ontology, which facilitates the matching. The process of matching is described as follow: firstly, search the common ancestor C of two nodes $c1$ and $c2$. In fact, C is a mapping pair$(c1', c2')$ got from the matching results of literal concepts, where $c1'$ is the ancestral node of $c1$ and $c2'$ is the ancestral node of $c2$. Secondly, calculate the semantic distance between $c1$ and $c2$ through C. Thirdly, do iterative calculation that search the common ancestor C of $c1'$ and $c2'$ until C is the virtual common node. Finally, add depth and density of nodes into the calculation. The formula is:

$$Sim\_struct(ID1, ID2) = Sim\big(Com\_ancestor(c1, c2)\big) \times \frac{\alpha k}{k + Dis(c1, c2)} + \beta\big(\eta + (1-\eta) \times \frac{e(c1) + e(c2)}{2}\big) + \frac{\gamma}{2}\big(\frac{d(c1)}{d(c1)+1} + \frac{d(c2)}{d(c2)+1}\big)$$

Where $Com\_ancestor(c1, c2)$ returns the common ancestor pair of $c1$ and $c2$, and $Dis(c1, c2)$ is the semantic distance, $e(c1)$ and $d(c1)$ are the density and depth of node[7]. The parameters $k(k > 0)$, $\eta(0 < \eta < 1)$, α, β and γ $(\alpha + \beta + \gamma = 1)$ control how much semantic distance, depth, density contribute to the calculation of structural similarity respectively.

$$Dis(c1, c2) = \sum_{x \in pn(c1)} wt\big(x, p(x)\big) + \sum_{x \in pn(c2)} wt\big(p(x), x\big)$$

$$wt(c, x) = Ls(c, x) \times T(c, x)$$

$$Ls(c, x) = -\log\big(P(c|x)\big) = -\log\frac{P(c \bigcap x)}{P(x)} = IC(c) - IC(x)$$

Where $wt(c, x)$ is the weight of $edge(c, x)$, $pn(c)$ is the set of nodes which are on the path from node c to the common ancestor node, $p(x)$ is the parent node of x, $IC(x)$is interest degree[6], $Ls(c, x)$ is the difference of the information content values between a child node and its parent, and $T(c, x)$ is the link relation factor.

There is something important to pay attention to, which makes the algorithm more efficiency:

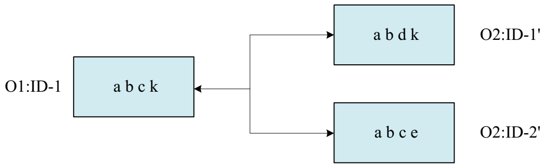


Fig. 3. Literal concept mapping of one to many

- This approach searches all the ancestor nodes of two nodes to be matched, and select the best matching path. If only search the nearest common ancestor node, the result may be wrong. For example: owing to the situation of one to many mappings in the matching results of literal concepts, it may occur that the mappings $(O1 : ID-2, O2 : ID-2')$ and $(O1 : ID-2, O2 : ID-4')$ got from results of literal concept matching are candidates for structural matching, but in fact $(O1 : ID-2, O2 : ID-2')$ is the best mapping. When comparing the node $O1 : ID-6$ and node $O2 : ID-6'$, if only search their nearest common ancestor, we will get a pair of nodes, $O1 : ID-2$ and $O2 : ID-4$. However, it is not the best mapping pair (we have known that the pair of $O1 : ID-2$ and $O2 : ID-2'$ is the best). To avoid this, we need to traverse all the common ancestors of nodes rather than the nearest. Then compare the iterative results and choose the best.
- Involve the literal interest degree. For instance, when we find mapping pairs $(O1 : ID-1, O2 : ID-1')$ and $(O1 : ID-1, O2 : ID-2')$ have the same structural similarity, and the values of their literal similarity calculations are both $3/4$ as shown in fig. 3, where a, b, c, d, e and k are Synsets, then the literal interest degree is needed to judge which the better matching object of $O1 : ID-1$ from $O2 : ID-1'$ and $O2 : ID-2'$ is: the less frequency of a Synset occurs in the ontology is, the more it contributes to the meaning of the node. So we calculate all the literal interest degrees of the common Synsets in each mapping pair using the formula metioned in Definition 4. And compare the maximal literal interest degrees of all the mapping pairs, then the max is the best matching because they contain the common Synset whose meaning is closer to concept of the phrase. To suppose the maximal literal interest degree of $(O1 : ID-1, O2 : ID-1')$ is $n1$ got from k, simultaneously, the maximal literal interest degree of $(O1 : ID-1, O2 : ID-1')$ is $n2$ got from e, and $n1 > n2$, we can draw the conclusion: $(O1 : ID-1, O2 : ID-1')$ should be the best mapping pair because $O1 : ID-1$ is more interested in Synset k.
- At last calculate the factors of density and depth of node. Because in each iteration the value of semantic distance should be multiplied by similarity of the common ancestor node which is smaller than 1, it will surely lead to the similarity of child nodes smaller than those of their ancestor nodes. This is contradictory to the role of depth and density calculation, because the nodes which have greater values of depth and density will have the larger value of similarity. Therefore, we must calculate the depth and density of node out of the procedure of calculation of semantic distance and iterations.

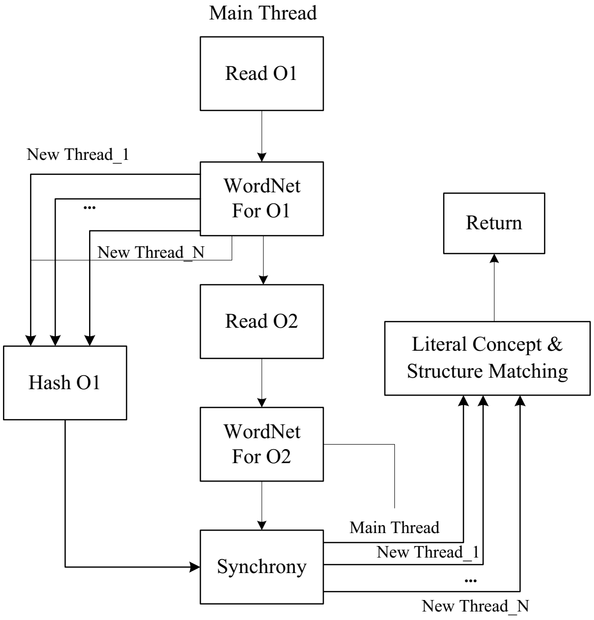


Fig. 4. Parallelization of the algorithm implemented by multi-threads

Parallelization of the algorithm NBJLM uses parallel algorithm to accelerate the process of matching. Fig. 4 shows the task partitioning. Firstly, we use the main thread to read $O1$ file and then look up the Synsets of all the node IDs of $O1$ in the WordNet. The reason of use only one thread is that this stage contains only IO operations which can not benefit from parallel execution and WordNet does not provide thread-safe APIs. Secondly, another multi-threads are launched to calculate hash values of node IDs' Synsets of $O1$, meanwhile we use the main thread to read $O2$ file and look up the Synsets of all the node IDs of $O2$. And these tasks could be run in parallel because one part is CPU operation, and another is IO operation. Finally, we synchronize all the threads, and then use them to calculate the literal concepts similarity and the structure similarity.

1.3 Adaptations made for the evaluation

This year, NBJLM has first taken part in OAEI. Therefore, in OAEI 2010 NBJLM used the match to compute the alignments for one track(anatomy). In order to assure the matching process is fully automated, all parameters are configured automatically with a strategy. No specific adaptations have been made.

1.4 Link to the system and parameters file

The version of NBJLM for OAEI 2010 can be downloaded from our website: $http://www.brsbox.com/OAEI2010$. The parameter file is also included in the NBJLM.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)

NBJLM alignment results for OAEI can be found at
$http : //www.brsbox.com/OAEI2010$.

2 Results

In this section, we describe the results of NBJLM algorithm against the Anatomy ontologies provided by the OAEI 2010 campaign. 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. This year we have participated in task#1 for the first time. Experiments were done on a computer with 1.8GHz AMDAthlon dual-core CPU and 2GB DDR2 RAM memory.

2.1 anatomy

Subtrack#1 In this subtrack, participants are asked to maximize F-measure. NBJLM used a threshold equal to 0.8 and obtained an F-measure equal to 85.8%. NBJLM obtained precision equal to 92.0% and recall equal to 80.3%. The runtime was 2 minutes.

3 General comments

3.1 Comments on the results

- **Strengths** NBJLM 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, NBJLM can achieve satisfactory alignments. And the way of minimizing the comparisons between entities, which leads to enhance running efficiency.
- **Weaknesses** NBJLM depends on the literal concept results to calculate structural similarity. So if the literals of concept missed, NBJLM will get bad results.

3.2 Discussions on the way to improve the proposed system

1) To enrich the semantic dictionaries because WordNet which is not a professional dictionary cannot obtain more comprehensive semantic concepts.
2) To take into account all concepts properties instead of only the hierarchicals ones.

4 Conclusion

This paper reports our first participation in OAEI campaign. We present the alignment process of NBJLM and describe the specific techniques for ontology matching. The method based on heterogeneous ontologies combines the calculations of literal concept and ontology structure and pays more attention to computational efficiency. 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. E. Agirre and G. Rigau. A proposal for word sense disambiguation using conceptual distance. *AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4*, pages 161–172, 1997.
2. Y. Jean-Mary, E. Shironoshita, and M. Kabuka. Ontology matching with semantic verification. *Web Semantics: Science, Services and Agents on the World Wide Web*, page 235C251, 2009.
3. Y. Li, D. McLean, Z. Bandar, J. O'Shea, and K. Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150, 2006.
4. M. A. Q. Muhammad Fahad. Similarity Computation by Ontology Merging System: DKP-OM. *Computer, Control and Communication*, pages 17–18, February 2009.
5. P. Resnik et al. Using information content to evaluate semantic similarity in a taxonomy. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 448–453. Citeseer, 1995.
6. S. Ross. A first course in probability. *New York*, 1994.
7. J. Sevilla, V. Segura, A. Podhorski, E. Guruceaga, J. Mato, L. Martinez-Cruz, F. Corrales, and A. Rubio. Correlation between gene expression and GO semantic similarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 2(4):338, 2005.
8. J. Tang, J. Li, B. Liang, X. Huang, Y. Li, and K. Wang. Using Bayesian decision for ontology mapping. *Journal of Web Semantics: Science, Services and Agents on the WorldWideWeb*, pages 243–262, 2006.

RiMOM Results for OAEI 2010

Zhichun Wang[1], Xiao Zhang[1], Lei Hou[1],
Yue Zhao[2], Juanzi Li[1], Yu Qi[3] , Jie Tang[1]

[1]Tsinghua University, Beijing, China
{zcwang,zhangxiao,greener,ljz,tangjie}@keg.cs.tsinghua.edu.cn
[2]Beihang University, Beijing, China
zhaoy1030@gmail.com
[3]National University of Defense Technology, Changsha, China
qiyu_418@sina.com

Abstract. This paper presents the results of RiMOM in the Ontology Alignment Evaluation Initiative (OAEI) 2010. We participate in three tracks of the campaign: Benchmark, IM@OAEI2010 (IMEI), and Very Large Crosslingual Resources (VLCR). We first describe the basic alignment process and alignment strategies in RiMOM, and then we present specific techniques used for different tracks. At last we give some comments on our results and discuss some future work on RiMOM.

1 Presentation of the system

Recently, ontology alignment has been developed as a key technology to solve interoperability problems across heterogonous data sources. Many automatic ontology alignment systems have been proposed and achieve good performance in real world data. With the development of Linked Data [1] and various social network websites, huge amount of semantic data is published on the web, which not only poses new challenges over traditional schema level ontology alignment algorithms, but also demands new techniques for instance matching.

RiMOM is a multistrategy dynamic ontology alignment system [2]. It implements several different matching strategies which are defined based on different ontological information. For each individual matching task, RiMOM can automatically and dynamically combine multiple strategies to generate a composed matching result. Recently, some new features were added into the new version of RiMOM which enable it to deal with unbalanced ontology matching [3], user interactive ontology matching [4], and large scale instance matching.

1.1 State, purpose, general statement

Currently, RiMOM is developed with a flexible framework for ontology alignment, where different kinds of alignment strategies can be plugged and configured easily. Fig 1 shows the architecture of RiMOM system.

The whole system consists of three layers: interface layer, task layer and component layer. In the interface layer, RiMOM provides a graphical user interface to allow users to customize the matching procedure: including selecting preferred components, setting the parameters for the system, etc. In semi-automatic ontology matching, user can also get involved in the matching process via the user interface. The task layer stores parameters of the alignment tasks, and controls the execution process of components in the component layer. In component layer, we define five groups of executable components, including preprocessor, matcher, aggregator, postprocessor and evaluator. In each group, there are several instantiated components. For a certain alignment task, user can select appropriate components and execute them in desired sequence.

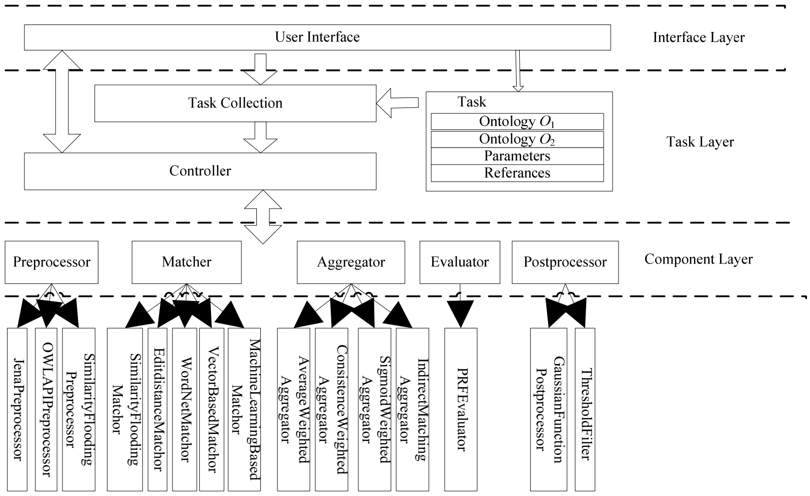


Fig 1. Architecture of RiMOM system

1.2 Specific techniques used

This year we participate in three tracks of the campaign: Benchmark, IM@OAEI2010 (IMEI), and Very Large Crosslingual Resources (VLCR). We describe specific techniques used in different tracks as follows:

Benchmark Track

For benchmark track, we use three matching strategies:

(1) Name based strategy: In this strategy, we calculate the edit distance between labels of two entities. Edit distance estimates the number of operation needed to convert one string into another. We define $(1-\#op/max\_length(l_1,l_2))$ as the similarity of two labels, where $\#op$ indicates the number of operations, $max\_length(l_1,l_2)$ represents the maximal length of the two labels.

(2) Metadata based strategy: In this strategy, we treat the information of each entity as a document, which consists of words in entity's label and comment. Then we construct a weighted feature vector using tf-idf technology, the similarity between two entities is then calculated as the cosine of the two vectors.

(3) Instance based strategy: In this strategy, we also construct a document for each entity, but the words are from the instances related to that entity. For a class entity, words in the label, comment and property value of all its instances are extracted as the entity's document; for a property entity, all the values it occurs in instances are extracted as the entity's document. Then the similarity between two entities is calculated as in Metadata based strategy.

When combining the results of different matching strategies, we use a different method from which we used in OAEI 2008 and 2009. Instead of aggregating similarity values before extracting final alignment, we first extract alignment based on each individual strategy by threshold filtering method, and then combine alignments of different strategies together. A similarity propagation procedure based on structure information is performed to find more mappings. The similarity propagation procedure is implemented in iteration; in each iteration, the similarity is propagated from already found mappings to the rest candidate mappings, candidate mappings which get high similarity are then added to found mappings; this process is repeated until no more mapping is found. This combination method can generate alignments with very high precision with acceptable recall.

Data Interlinking track

The DI (Data Interlinking) track is designed to test the ontology matching systems' ability on link generation of LinkedData. There are five datasets, i.e. DailyMed, Diseasome, DrugBank, Sider and LinkedMDB, requested to be matched to related datasets in the LinkedData respectively. These data sets are all comes from the real world data and in relatively larger scale than the generated dataset. We choose four datasets in the domain of medicine to test our algorithm while exclude the linkedMDB dataset. According to our observations on the instance data, we split the information in the instance into six categories: the URL, the Meta Information, the Name, the string type information, the non-string type information and the neighboring information. Among the six categories the Name, which usually comes from the rdfs:label property or other ontology specific property such as foaf:name) is the most distinguishing feature to identify an instance. In addition, the natural language information and the neighboring instances are very useful, too. Thus we propose a vector based method for the DI track. We build two vectors referred to as Name Vector and Virtual Document for each instance. The Name Vector is constructed by accumulating the terms in the Name property values and setting the occurrence of each term as its weight. For Virtual Document, we first collect the terms of the each instance's descriptions and annotations then fetch the local information of its neighboring instances to construct a comprehensive vector. Because the Virtual Document Space is much larger, we compute the tf-idf value of each term as its weight. The similarity between two instances is calculated as the weighted sum of their similarity (Cosine Distance) on two kinds of vectors respectively. However, this method is infeasible on large scale input because pair-wise comparisons on instances are too costly. Thus we introduce a candidate selection process. Only the instance pairs which are selected as candidate mappings are compared. Generally we use two rules for candidate selection: 1) instances with common terms in their Name Vectors; 2) instances with common top weighted terms in their Virtual Documents. To utilize the functionality, we build inverted index of instances for terms in Name Vector and top weighted terms in the Virtual Document. Consequently our algorithm can generate the candidates very quickly and eliminate the meaningless comparisons between unrelated instances. Several experiment results show that the candidate selection will not eliminate the possible alignments in most of the cases. In the following phase of the algorithm, we may use the Meta Information and non-string type values as restrictions to filter the results according to the instance characteristics. For example, a common one is that those instances whose classes are not matched will be filtered out. At last a threshold is used on similarity for the final result. Totally speaking, this method is a generic and efficient method for instance matching.

IIMB and PR track

Traditionally, information of individuals in an ontology is frequently utilized in supporting of schema matching. Inversely, information of schema is of equal importance in alignment of individuals that are sharing the same ontology structure. Thus, for the Instance Matching Track of this year, we take more about schema information, especially classes and properties, into consideration in aligning individuals.

For Instance Matching, our main idea is that we classify individuals by their classes, complete information of each individual as complete as possible, run matching algorithm for each class respectively, and compute similarity of two candidates based on weight-mean of properties assigned with specified weights. And the algorithm can be generalized as four consecutive phase:

Preprocessing: Read and store the schema information for further use. Build a local schema that connects properties and classes and implement it by learning information of individuals.

Information Complementation: Modify the information of each individual, aiming at making them as complete as possible. We defined some rules for judging the validity of values, as well as for solving the transformations in value, structure and logical. Reclassify individuals by recognizing and comparing properties they carry with those in classes, based on our local schema implemented in the previous phase.

Matching: Given the facts that different properties of individuals play quite different roles, and that every individual has its unique characteristic(s), for each property, we assign it with a specified weight and combine this weight with string-based similarity value computed under Edit Distance or Vector based algorithm. We assign the weight-mean of properties as the final similarity value.

Spread Similarity: In order to fully utilize the connection of individuals, we apply a similarity-flooding-like algorithm to spread the similarity.

1.3 Adaptations made for the evaluation

In order to deal with large scale data sets, we use an inverted index technique to accelerate the speed of locating and reading data.

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 of RiMOM for OAEI 2010 Campaign are available at http://keg.cs.tsinghua.edu.cn/project/RiMOM/OAEI2010/

2 Results

As introduced above, RiMOM participates in three tracks in OAEI 2010; we present the results and related analysis below.

2.1 Benchmark

There are 111 alignment tasks in benchmark data set; we divide these tasks into three groups: 1xx, 2xx, and 3xx. We compare the results of RiMOM in OAEI 2010 and OAEI 2009 [5] in Table 1. It can be observed that the performance of RiMOM in 1xx task continues to be perfect as last year; as for the 2xx task, the result of this year is better than that of last year, with regard to both precision and recall; the precision of 3xx increases this year, but the recall decreases, while the F1-measure is almost the same as last year. Overall, the precision, recall and F1-measure for the entire benchmark data set of RiMOM this year achieve 99% precision, 84% recall and an F1-measure of 91%. Compared with last year's result, there are 6% improvement on precision, 2% improvement on recall and 4% improvement on F1-measure.

Table 1. Benchmark test results of RiMOM in OAEI 2010 and OAEI 2009
(Values are real precision and recall and not an average of precision and recall)

Test	OAEI 2010			OAEI 2009		
	Prec.	Rec.	F1	Prec.	Rec.	F1
1xx	1.00	1.00	1.00	1.00	1.00	1.00
2xx	0.99	0.83	0.91	0.93	0.81	0.87
3xx	0.91	0.74	0.82	0.81	0.82	0.81
H-mean	0.99	0.84	0.91	0.93	0.82	0.87

2.2 DI track of IM@OAEI2010

We generate results for four of five datasets in the track except the LinkedMDB dataset. Since we are requested to mapping each dataset to several related datasets in LinkedData and these datasets not provided in the track, we download these datasets and transfer them into RDF format using Jena. As a result we cannot get some datasets such as STITCH because there is only a SPARQL endpoint for it. We also found there are many duplicate entries in the reference alignment of Sider and the namespace for DBpedia in the reference alignment of Drugbank is not uniform, we adjust these reference files to get the final result of our algorithm. We set the parameter of our algorithm as NameWeight = 0.6 and threshold = 0.55. The result of

Sider dataset is shown in Table 2. From the result we can see that according to the different characteristics of the instance file, the results may be very different: some are high in precision and some are high in recall. For those high in recall but low in precision, more careful filter may be added to the algorithm by studying the data. On the other hand, for those low in recall, the threshold may be cut down.

Table 2. The result of Sider Dataset

DataSet	DBpedia	DailyMed	Diseasome	Drugbank	TCM	STITCH	TOTAL
Precision	0.717	0.567	0.315	0.961	0.778	/	0.617
Recall	0.482	0.706	0.837	0.342	0.812	/	0.467
F-Measure	0.576	0.629	0.458	0.504	0.795	/	0.532

The result of DailyMed dataset is shown in Table 3. The result of our algorithm is extremely bad in the LinkedCT dataset. It generates a lot of results (up to 100,000) so that the precision is very low. Because of the dominance of LinkedCT results in the reference, our result in total is not good, too. According to our observation on the reference alignment of LinkedCT, they are automatically generated from the owl:SeeAlso property in the file. After reviewing some of our results, we found that many of our results are reasonable but some of the references are not, we think the reference alignment is not very complete and sound. However, our algorithm cannot generate good results from DBpedia means we need much more improvement on it. The other two datasets with LinkedCT reference, Diseasome and DrugBank are similar in results.

Table 3. The result of DailyMed dataset

DataSet	DBpedia	LinkedCT	TCM	Sider	TOTAL
Precision	0.246	0.070	0.159	0.567	0.085
Recall	0.293	0.235	0.535	0.706	0.296
F-Measure	0.267	0.107	0.123	0.629	0.132

Table 4. Results of IIMB

IIMB_SMALL				**IIMB_LARGE**			
Dataset	**Prec.**	**Rec.**	**F1**	**Dataset**	**Prec.**	**Rec.**	**F1**
001 - 020	0.975	0.975	0.975	**001 - 020**	0.997	0.994	0.995
021 - 030	0.861	0.710	0.778	**021 - 030**	0.798	0.696	0.744
031 - 060	0.913	0.953	0.933	**031 - 040**	0.843	0.766	0.803
061 - 070	0.809	0.639	0.714	**041 - 060**	0.877	0.976	0.924
071 - 080	0.792	0.500	0.613	**061 - 070**	0.663	0.586	0.622
				071 - 080	0.575	0.557	0.566

2.3 IIMB track of IM@OAEI2010

The result for IIMB_SMALL and IIMB_LARGE is shown in Table 4. As the number of datasets increases, the text-based information the dataset contains decrease while complex combination of modifications increase, thus the performance of our algorithm decreases since it is anyway fundamentally based on string comparison. We can also see that with the amount of instances grows, the influences brought by the noise increase, which do nothing but harm to effect of our algorithm.

2.4 PR track of IM@OAEI2010

PR track consists of three subtasks; the results for these tasks are shown in Table 5. It can be observed that RiMOM gets perfect performance on the first task; for the second task, RiMOM gets really good recall and the precision is 95.2%; for the last task, the precision and recall both decrease compared to the former two tasks.

Table 5. Results of PR

Dataset	Precision	Recall	F-Measure
Person11 - Person12	1.0	1.0	1.000
Person11 - Person12	0.952	0.99	0.971
Restaurant1 - Restaurant2	0.86	0.768	0.811

2.5 VLCR track

The purpose of VLCR task is to match three resources to each other, namely, the Thesaurus of the Netherlands Institute for Sound and Vision (called GTAA), the New York Times subject headings and DBpedia. Each resource consists of lots of instances: 142,000 in GTAA, 12,000 in NYT and 7,500,000 in DBpedia. Table 6 lists the number of the mapping we found.

Table 6. Result for VLCR task

Dataset		Number of mappings
NYT-DBpedia		9257
GTAA-DBpedia		68337
NYT-GTAA	Direct mapping	4324
	Indirect mapping	4487

Due to the lack of information, sometimes it is very difficult to match two instances in NYT and GTAA directly. Since we have mapped the two relatively small instance sets to DBpedia, it is possible to use the map results to get more maps between the two small one. Instances in NYT and GTAA matches to the same instance in DBpedia will be added to the final results.As shown in the table, NYT–DBpedia, GTAA-DBpedia and NYT-GTAA are three subtasks of VLCR task. Indirect matching find 163(rise by 3.7%) new mappings in NYT-GTAA task.

3 General comments

By far instance matching, especially matching on real world instance is still a very challenging problem. Instance Matching is of great importance for bringing the ontology matching into practical use with its wide range of application scenarios. Instance matching shows its special characteristics compared with the conventional schema matching and the large scale nature of instance matching is a big obstacle to employ the existing methods. A relatively generic and efficient method for instance matching is in great need. The IMEI track of OAEI 2010 provides a good platform to test the instance matching algorithms and this area will attract more attention in the community.

4 Conclusion

In this paper, we present the results of RiMOM in OAEI 2010 Campaign. We participate in three tracks this year, including Benchmark, IMEI, and VLCR. We have presented the architecture of RiMOM system and described specific techniques used in this campaign. In this campaign, we design a new strategy combination method for benchmark tracks, and get better performance than last year. We particularly focus on the instance matching task; propose some new strategies for these tasks. The results illustrates that our system RiMOM can achieve good performance in both schema matching and instance matching tracks.

Acknowledgement:

The work is supported by the National Natural Science Foundation of China (No. 60973102), the National Basic Research Program of China (973 Program) (No. 2007CB310803), the National High-tech R&D Program (No. 2009AA01Z138), it is also supported by IBM SUR joint project.

References

1. http://linkeddata.org/.
2. 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.
3. 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.
4. 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.
5. 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.

Alignment Results of SOBOM for OAEI 2010

Peigang Xu, Yadong Wang, Liang Cheng, Tianyi Zang
School of Computer Science and Technology
Harbin Institute of Technology, Harbin, China
peigang.xu@gmail.com, ydwang@hit.edu.cn, chl198478@126.com,tianyi.zang@gmail.com

Abstract. In this paper we give a brief explanation of how Sub-Ontology based Ontology Matching (SOBOM) method gets the alignment results at OAEI2010. SOBOM deal with an 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 anchors provided by a linguistic matcher. And then it extracts sub-ontologies based on the anchors and ranks these sub-ontologies according to their depths. Secondly, SOBOM utilizes Semantic Inductive Similarity Flooding algorithm to compute the similarity of concepts between different 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.

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 Sub-ontology based Ontology Matching (SOBOM) is designed for matching light-weight ontologies that has is-a hierarchy as their backbones. It matches 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 developed to match ontology automatically for general purpose. Based on two different views, we design three elementary matchers in current version. The first one is a anchor generator which is used to find anchors; the second one is a structure matcher SISF (Semantic Inductive Similarity Flooding) which is inspired by Anchor-Prompt [3] and SF [4] algorithms and is exploited to flood similarity among concepts. The last one is a relationship 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 anchors got by linguistic matcher and rank them by their depths descendingly. Overall SOBOM is a sequential method, so it does not care how to combine the results of different matchers. The overview of the method is illustrated in Fig. 2.

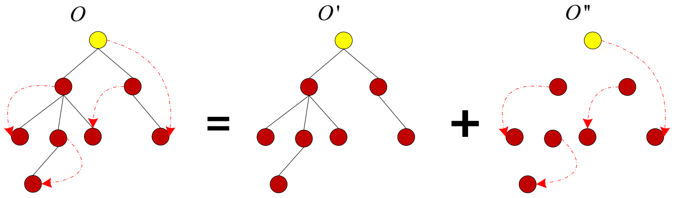


Fig. 1. Two views of an ontology in SOBOM

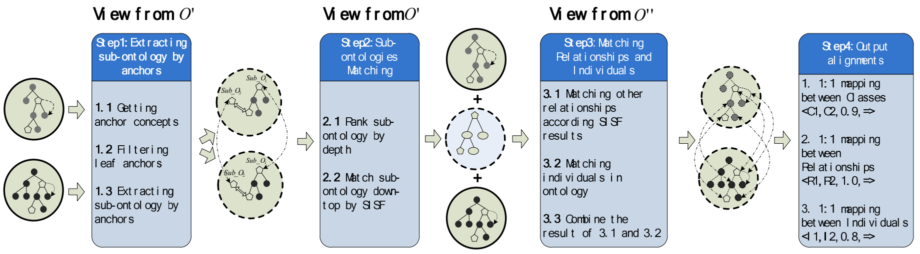


Fig. 2. The processing overview of SOBOM

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. 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:

$$Domain\ (r) = \{c_i \mid c_i \in C\ \ and\ \ having\ \ the\ \ relationsh\ \ ip\ \ r\}$$

$$Range\ (r) = \{c_i \mid c_i \in C\ \ and\ \ can\ be\ \ value\ \ of\ \ r\}$$

Anchor: An anchor is defined as a pair of assumed equivalent non-leaf concepts across ontologies. Given two ontologies O_1, O_2, $c_1 \in O_1, c_2 \in O_2$, if $c_1 \equiv c_2$ (means that c1 is identical with c2),and c1, c2 are both not leaf nodes in the hierarchies of O1 and O2 respectively, then an anchor, X is defined as a pair of concepts $< c_1, c_2 >$.

Sub-Ontology: Let ontology $O = (C,R, H^C,H^R,)$. A sub-ontology O^x is a subset of O whose elements all come from O, called $O^x = (C_1,H_1^C)$, where $C_1 \subseteq C, H_1^C \subseteq H^C$, x is the root of H^C. Indeed, a sub-ontology in our method is a hierarchical taxonomy, and its root is an anchor concept.

Sub-ontology Depth. *The depth of sub-ontology O^x is the maximal length of path from the anchor x to the root r_i of the taxonomy H_i^C which contains it in the original ontology O.*

$$Depth(O^x) = Max(|x \to ..., \to r_i|),\ r_i \in H_i^C,\ H_i^C \in O$$

1.2 Specific techniques used

SOBOM aims to provide high quality of 1:1 alignments between concept and property pairs. We have implemented SOBOM algorithm in java and integrated three distinguishing constitutional matchers. They are 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].

- Our anchor generator is based on the local context of an entity in ontology. In details, the local context of an entity including the following aspects: the textual information (label, id, comments and so on), the structure information (the number of super or sub concepts, the number of constraints) and the individual information (the number of individual if existing). We consider that the local context of an entity can express the meaning of it. Consequently, we get three similarity matrixes respectively, and we choose the maximal of them as the final results.

- SISF uses the RDF statement to represent the ontology and utilizes the anchors to inducting the construction of similarity propagation graph for the sub-ontologies. SISF handles 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 SOBOM and extracts sub-ontologies according to the anchors [5,6]. SoE ranks extracted sub-ontologies according to their depths. As we extract sub-ontologies for ontology matching, the rules of extracting sub-ontology in SoE are as following: only sub-concepts of anchor are included in the sub-ontology. In other words, a sub-ontology is a taxonomy which has anchor as root.

If one of the two concepts in an anchor is a leaf node in the original ontology, we do not use SISF to deal with it actually. Because this phenomenon just represents a one-to-many mapping. 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 2010 campaign. All the alignments outputted by SOBOM are based on the same set of parameters.

1.4 Link to the system and parameters file

The current version of SOBOM is available at: http://mlg.hit.edu.cn:8080/Ontology/Download.jsp, and the parameters setting is illustrated in the reading me file.

1.5 Link to the set of provided alignments (in align format)

We deploy our matcher as a web service, our web service name is: eu.sealsproject.omt.ws.matcher.AlignmentWSImpl. The endpoint of our web service can be found at: http://mlg.hit.edu.cn:8080/SOBOMService/SOBOMMatcher?wsdl.

2 Results

In this section, we describe the results of SOBOM algorithm against the benchmark, directory and anatomy ontologies provided by the OAEI 2010 campaign. We use Jena-API to parse the RDF and OWL files. The experiments were carried out on a PC running Windows vista ultimate with Core 2 Duo processors 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. SOBOM is a sequential matcher. If the linguistic matcher gets no results, SOBOM will produce no result. 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, their structures are quite similar. SOBOM is a sequential matcher, our anchor generator matches concepts based on their local context not only the linguistic information. So, although without linguistic information SOBOM also gets relatively high precision and 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 anchor generator, and both the precision and recall are pretty good.

#248-266 Both the linguistic and structural characteristics of the candidate ontologies are changed heavily, so the tests in this group might be the most difficult ones in all the benchmark tests. So, SOBOM does not very well.

#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
Precision	1.0	0.99	0.99	0.87	0.77
Recall	1.0	0.85	0.99	0.57	0.70

Compared to our previous results (OAEI2009), the recall of every group is highly improved. This is enhanced by our redesigned anchor generator.

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). These are relatively large compared to benchmark ontologies. This type ontologies is what SOBOM suitable for handling, it generated 268 sub-ontologies and 1249 alignments between MA and NCI, consumed 19min3s to complete the matching task. It is obvious that most of the alignment appears in the leaf nodes in ontologies (834 leaf node alignments). The experiment result shows in Table 2.

Table 2. The performance of SOBOM on the anatomy test

	Leaf node alignments	Sub-ontologies	Total Alignments	Time consuming

NCI	834	268	1249	19min3s
MA				

2.3 Conference

There are 120 pairs of ontologies in this track. Most of them are blind tests (i.e. there no reference alignment available). The whole results are available at: http://seals.inrialpes.fr/platform/;jsessionid=FD1E3A5CE8DA43C1D52DB21079EAECF3?wicket:bookmarkablePage=:eu.sealsproject.omt.ui.Results&endpoint=http://219.217.238.162:8080/SOBOMService/SOBOMMatcher?wsdl&evaluationID=http://219.217.238.162:8080/SOBOMService/SOBOMMatcher?wsdl2010/10/03+02:09:03&track=Conference+Testsuite.

3 General comments

In this section, we want to introduce comments on the results of SOBOM algorithm and the way to improve it.

3.1 Comments on the results

Strengths SOBOM deals with ontology from two different views and combines results of every step in a sequential way. If the ontologies have regular literals and hierarchical structures, SOBOM can achieve satisfactory alignments. And it can avoid missing alignment in many partitioning matching methods as illustrated in [7].

Weaknesses SOBOM needs anchors to extract sub-ontologies. So it depends on the precision of anchors. In current version, we use a linguistic matcher to get anchor concept, 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, we have enhanced the anchor generator by not considering the textual information but also the structure information. Our next plan is to integrate a more powerful matcher to produce anchor concepts or develop a new method to get anchor concepts. Meanwhile, we plan to develop a mapping debugging method to refine the results of SOBOM.

4 Conclusion

Ontology matching is very important part of establishing interoperability among semantic applications. This paper reports our participation in OAEI2010 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. The unique feature of our method is combining sub-ontology extraction with ontology matching based on two different views of an ontology.

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, H.J. Tao: SOBOM: Sub-ontology based Ontology Matching. To be appear.

TaxoMap alignment and refinement modules: Results for OAEI 2010

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] The University of Memphis
Memphis, TN, USA
`nb.niraula@gmail.com`

Abstract. *TaxoMap* is an alignment tool which aims to discover rich correspondences between concepts (equivalence relations (*isEq*), subsumption relations (*isA*) and their inverse (*isMoreGnl*) or proximity relations (*isClose*)). 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* uses a pattern-based approach implemented in the *TaxoMap Framework* helping an engineer to refine mappings to take into account specific conventions used in ontologies.

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. Then, to find mappings we use the following available elements: labels of concepts and hierarchical structures.

The new implementation of *TaxoMap* introduces a step of refinement of mappings (the alignment results) which extends the alignment process and completes it.

We take part to three tests. We hope the new step of refinement helps us to perform better in terms of precision of generated mappings.

2 Presentation of the System

Our system is composed of two elements: *TaxoMap*, the alignment tool and *TaxoMap Framework*, an environment allowing to specify and perform refinement treatments applied on the prior obtained mappings.

2.1 State, Purpose and General Statement

TaxoMap has been designed to align owl ontologies $O = (C, H)$. C is a set of concepts characterized by a set of labels and H is a subsumption hierarchy which contains a set of *isA* relationships between nodes corresponding to concepts. The alignment process is an oriented process which tries to connect the concepts of a source ontology O_S to the concepts of a target ontology O_T. The correspondences found are equivalence relations (*isEq*), subsumption relations (*isA*) and their inverse (*isMoreGnl*) or proximity relations (*isClose*).

To identify these correspondences, *TaxoMap* implements techniques which exploit the labels of the concepts and the subsumption links that connect the concepts in the hierarchy. The morpho-syntactic analysis tool, *TreeTagger* [1], is used to classify the words of the labels of the concepts and to divide them into two classes, *full words* and *complementary words*, according to their category and their position in the labels. At first the repartition between *full* and *complementary words* is used by a similarity measure that compares the tri-grams of the labels of the concepts [2] and gives more weight to the common *full words*. Then it is also used by the alignment techniques. For example, one technique named $LabelInclusion$ generates an *isA* mapping between c_s and c_{tmax} if (1) the concept c_{tmax} is the concept of O_T having the highest similarity value with the concept c_s of O_S, (2) one of the labels of c_{tmax} is included in one of the labels of c_s, (3) all the words of the included label of c_{tmax} are classified as *full words* by *TreeTagger*.

Given a concept c_S of the ontology source O_S, our similarity measure identify the concept c_{tmax} of the target ontology O_T which have the highest similarity with c_S. The alignment techniques are then used to decide if the concept c_S can be effectively aligned with this concept c_{tmax} and which relation should be established between the two concepts, or whether, another concept of O_T must be chosen. A proposed mapping belongs to a single method, a concept of O_S can be aligned at most with one concept of O_T. In contrast, the concepts of O_T may be involved in several proposed alignments.

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_{tmax} and one label of c_s is greater than a threshold (Equiv.threshold).
- 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 the relationships $< c_s\ isMoreGnl\ c_{tmax} >$.
- High lexical similarity: If the similarity measure of c_{tmax} is greater than a threshold (HighSim.threshold) and if one of its labels shares at least two full words in common with one of the labels of c_s, without being including in the labels of c_s, the heuristic generates the relationship $< c_s\ isClose\ c_{tmax} >$.

- 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 two following techniques can be used:
  - 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).
- Best similarity: If none of the above techniques is applicable, the relationship $< c_s\ isClose\ c_{tmax} >$ is generated if the similarity of c_{tmax} is greater than a threshold (Better.thresholdMax).
- Property similarity: Two classes c_s and c_t are likely to be aligned if they share the same properties. Our property similarity is computed using Degree of Commonality Coefficient presented in [8].

Mappings identified by *TaxoMap* are generated in the Alignment format used as a standard in the OAEI campaign. We added to this format the information about the names of the techniques that generated mappings. The aim is to facilitate the specification of treatments exploiting the mappings generated by those techniques. All these pieces of information are stored in a relational mappings database which can then be queried using *SQL* queries. This allows, in particular, to present the generated mappings to the expert in the validation phase, technique by technique.

In the OAEI campaigns, only equivalence relations are evaluated in the alignment contest. This has important implications on our results:

1. None of the mappings generated by the label inclusion techniques that lead to a subsumption relation isA is considered as such. Most of them are wrong if they are converted into equivalence relation.
2. Moreover, while it is natural to consider that several different concepts can be relied by a subsumption relation to the same concept, the conversion of subsumption relation into equivalence relation leads to the creation of multiple equivalence relations for a same concept.
3. As a concept of O_T must have only one equivalent concept in O_S in OAEI campaigns, if we consider the mappings leading to subsumption or proximity relations, all mappings which connect a concept of O_S to a concept of O_T which is already involved in an equivalence relation are false.

We will see in the next section how the *TaxoMap* refinement module [7] will allow us to remove these incorrect mappings.

2.2 TaxoMap Framework

We proposed an environment allowing to specify and perform treatments applied on the prior obtained mappings. At first, this environment will be used to improve the quality of an alignment provided by *TaxoMap*. Subsequently, it will be used for other treatments based on mappings as enriching, restructuring or merging ontologies.

An important feature of the approach is to allow a declarative specification of treatments based on particular alignment results, concerning particular ontologies and using a predefined vocabulary. Treatments which can be specified depend on the characteristics of the concerned ontologies and on the task to be performed (at first mapping refinement and subsequently ontology merging, restructuring, enriching). These treatments are thus associated to independent specification modules, one for each task, each having their own vocabulary. The approach is extensible and a priori applicable to any treatment based on alignment results.

We present the Mapping Refinement Pattern Language (MRPL) used to specify mapping refinement pattern. This language differs from the one defined in [3] especially because it includes patterns which test the existence of mappings generated by alignment techniques.

The vocabulary of MRPL contains:

- *a set of predicate constants*. We distinguish three categories of predicate constants: the predicate constants relating to the type of techniques applied in the identification of a mapping by TaxoMap, the predicate constants expressing structural relations between concepts of a same ontology, the predicate constants expressing terminological relations between labels of concepts.
- *a set of individual constants*: $\{a, b, c, ...\}$
- *a set of variables:* $\{x, y, z, ..., \_\}$ where $\_$ is an unnamed variable used to represent parameters which do not need to be precised.
- *a set of built-in predicates:* $\{Add\_Mapping$, $Delete\_Mapping\}$
- *a set of logical symbols*: $\{\exists$, $\wedge$, $\neg\}$

MRPL allows the definition of a **context part** which must be satisfied to make the execution of a pattern possible, and of a **solution part** which expresses the process to achieve when the **context part** is satisfied.

Context part of pattern

The **context part** tests (1) the technique used to identify the considered mapping, (2) the structural constraints on mapped elements, for example, the fact that they are related by a subsumption relation to concepts verifying or not some properties, or (3) the terminological constraints, for example, the fact that the labels of a concept are included in the labels of other concepts. These conditions are represented using formulae built from predicate symbols. So, we distinguish three kinds of formula according to the kind of predicate symbols used.

The formulae related to the type of techniques applied in the identification of a mapping by TaxoMap. By testing the existence in the mappings database of a particular relation generated by a given technique, we build formulae that implicitly test the conditions for the application of this technique. For example the formula $isAStrictInclusion(x, y)$ tests the existence of a mapping *isA* generated between two concepts x and y using the technique searching $LabelInclusion$, t_2. It validates implicitly at the same time all the conditions for the application of t_2, i.e. (1) the concept y is the concept of O_T having the highest similarity value with the concept x of O_S, (2) one of the labels of y is included in one of the labels of x, and (3) all the words of the labels of y are classified as *full words* by *TreeTagger*. TaxoMap includes several alignment techniques. Thus, several predicate symbols leading to formulae of that kind are needed. More formally, let:

$R_M = \{isEq, isA, isMoreGnl, isClose\}$, the set of correspondence relations used by TaxoMap,

$T = \{t_1, t_2, ...\}$, the set of techniques.

T_M, the table storing generated mappings in the form of *4-tuple* (x, y, r, t) where $x \in C_S, y \in C_T, r \in R_M, t \in T$. The pairs of variables (x, y) which can instantiate these formulae will take their values in the set $(x, y) \mid (x, y, r, t) \in T_M$. The predicate symbols necessary for the task of refinement presented in this paper are $isEquivalent$ and $isAStrictInclusion$ the semantics of which are the following:

- $isEquivalent(x, y)$ is true iff $\exists(x, y, isEq, t_1) \in T_M$
- $isAStrictInclusion(x, y)$ is true iff $\exists(x, y, isA, t_2) \in T_M$
- $mapping(x, y)$ is true iff $\exists(x, y, \_, \_) \in T_M$

The formulae expressing structural relations between concepts x and y of the same ontology $O = (C, H)$. Since the aim of *TaxoMap* is the alignment of taxonomies, the structural relations considered here are subsumption relations. If the approach was used with another alignment tool, other relations could be considered. Note that the instances of variables in these formulae will be constrained, either directly because they instantiate the previous formulae, related to the type of the applied techniques, or indirectly by having to be in relation with other instances.

- $isSubClassOf(x, y, O)$ is true $\Leftrightarrow isA(x, y) \in H$
- $isParentOf(x, y, O)$ is true $\Leftrightarrow isA(y, x) \in H$
- $conceptsDifferent(x, y)$ is true $\Leftrightarrow ID(x) \neq ID(y)$ with $ID(x)$ is the identifier of the concept x.

The formulae expressing terminological relations between the labels of the concepts: not detailed here because not used in the examples of this paper.

Solution part of pattern

A **context part** is associated to a **solution part** which is a set of actions to be performed. This set of actions is modeled by a conjunction of built-in predicates executed in a database. The built-in predicates are defined as follows:

- $Add\_Mapping(x, y, r)$ has the effect of adding a tuple to the table T_M which becomes $T_M \cup \{(x, y, r, t)\}$ where r and t are fixed in the treatment condition by instantiating the predicate corresponding to the type of technique associated with the considered mapping.
- $Delete\_Mapping(x, y, \_)$ has the effect of removing a tuple from the table T_M which becomes $T_M - \{(x, y, \_, \_)\}$.

Mapping Refinement Pattern used in OAEI

Pattern-1: This pattern concerns mappings generated by the technique t_1, connecting by an equivalence relationship a concept y of the target ontology O_T with a concept x of the source ontology O_S. Because a concept y of the target ontology O_T must be involved in at most one equivalence relation, mappings involving y and obtained from other techniques than t_1 should be removed.

Context part of Pattern-1:
$\exists x \exists y\ (isEquivalent(x, y)$
$\wedge\ \exists z\ (mapping(z, y) \wedge conceptDifferent(z, x)$))
Solution part of Pattern-1:
$Delete\_Mapping(z, y, \_)$

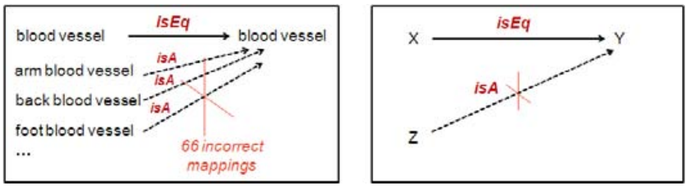


Fig. 1. Illustration of Pattern-1

Pattern-2: For the anatomy subtask 4, if we know a set of reference mappings, and we know that the pair (x, y) belongs to this set, we could express a new refinement pattern to remove generated mappings that relie to the concept y of the target ontology, concepts z of the source ontology other than x.

We should define the new predicate $referenceMapping(x, y)$ as follow:
$referenceMapping(x, y)$ is true iff $\exists (x, y) \in Set\_Reference\_Mapping$.

Context part of Pattern-2:
$\exists x \exists y\ (referenceMapping(x, y)$
$\wedge\ \exists z\ (mapping(z, y) \wedge conceptDifferent(z, x)$))
Solution part of Pattern-2:
$Delete\_Mapping(z, y, \_)$

2.3 Link to the system and parameters file

TaxoMap requires:

- Java (Version 1.6 and above)[3]

The version of *TaxoMap* used in 2010 contest can be downloaded from:

- http://www.lri.fr/ hamdi/TaxoMap/TaxoMap.html

---

[3] http://java.sun.com

2.4 Link to the Set of Provided Alignments

The alignments produced by *TaxoMap* are available at the following URLs:
http://www.lri.fr/˜hamdi/OAEI10/

3 Results

3.1 Benchmark Tests

Since our algorithm only provides mapping for concepts, the recall is low even for the reference alignment. The overall results show a slight improvement over those the 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. We considered *Human* as the target ontology as is it well structured and larger than *Mouse*. *TaxoMap* performs the alignment in about 12 minutes.

Table 1. Results of *TaxoMap* in the different tasks

	Task#1	Task#2	Task#3
# Computed Mappings	1 449	1 127	2 357
Precision without Refinement	0.779	0.929	0.477
# Removed Mappings	226	32	945
# Submitted Mappings	1 223	1 095	1 412
Precision	0.924	0.956	0.838
Δ Precision with Refinement	+ 0.145	+ 0.027	+ 0.361
Recall	0.743	0.689	0.774
F-measure	0.824	0.801	0.802

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 subtask 3) and we change *isClose* mapping to equivalence. In addition, we use the refinement pattern described above to delete mappings between a concept of the target ontology that was already aligned with an equivalence mapping. As a result, we found fewer mappings than last year but the precision is better [4].

3.3 Directory Test

The directory task consists of Web sites directories like Google, Yahoo! or Looksmart. Two modalities are proposed this year:

- Small tasks: includes 4,639 tests represented by pairs of OWL ontologies.
- Single task: includes only one matching task. The source and the target ontologies to be matched contain 2854 and 6555 concepts respectively.

TaxoMap takes about 40 minutes to complete each modality.

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.

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 hierarchical ones.
- To use 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 [5] [6].

5 Conclusion

This paper reports our participation to OAEI campaign with the new implementation of *TaxoMap*. Our participation in the campaign allows us to test the robustness of *TaxoMap*, the new implemented techniques and the efficacity of the new refinement module in *TaxoMap Framework*.

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] Scharffe, F.: Correspondence Patterns Representations. PhD thesis, Univ. of Innsbruck, 2009.

[4] Hamdi, F., Safar, B., Niraula, NB. and Reynaud, C. TaxoMap in the OAEI 2009 alignment contest, Proceedings of the ISWC'09 Workshop on Ontology Matching OM-09 (2009)

[5] Reynaud, C. and Safar, B. When usual structural alignment techniques don't apply, The ISWC'06 Workshop on Ontology matching (OM-06), (2006)

[6] Reynaud, C. and Safar, B. Exploiting WordNet as Background Knowledge,The ISWC'07 Workshop on Ontology Matching (OM-07), (2007)

[7] Hamdi, F., Reynaud, C. and Safar, B., Pattern-based Mapping Refinement, 17th International Conference on Knowledge Engineering and Knowledge Management, EKAW (2010)

[8] Reul, Q. and Jeff Z. Pan, KOSIMap: Ontology alignments results for OAEI 2009, Proceedings of the ISWC'09 Workshop on Ontology Matching OM-09 (2009)

Towards a UMLS-based silver standard for matching biomedical ontologies

E. Jiménez-Ruiz[1], B. Cuenca Grau[2], I. Horrocks[2], and R. Berlanga[1]

[1] Universitat Jaume I, Spain, `{ejimenez,berlanga}@uji.es`
[2] University of Oxford, UK, `{berg,ian.horrocks}@comlab.ox.ac.uk`

Abstract. We propose a silver standard based on the UMLS Metathesaurus to align NCI, FMA and SNOMED CT. This silver standard aims at being exploited within the OAEI and SEALS Campaigns.

1 Motivation

The UMLS Metathesaurus (UMLS-Meta) [1] is currently the most comprehensive effort for integrating independently-developed medical thesauri and ontologies. UMLS-Meta is being used in many applications, including *PubMed* and *ClinicalTrials.gov*. The integration of new UMLS-Meta sources combines automatic techniques, expert assessment, and auditing protocols (see [2] for a review of current methods). In its 2009AA version, UMLS-Meta integrates more than one hundred thesauri and ontologies, including SNOMED CT, FMA and NCI, and contains more than 6 million entities. UMLS-Meta provides a list with more than two million unique identifiers (CUIs). Each CUI can be associated to entities belonging to different sources. Pairs of entities from different sources with the same CUI are synonyms and hence can be represented as an equivalence mapping. Thus, UMLS-Meta mappings could be considered as a *silver standard* to align ontologies such as SNOMED CT, FMA or NCI. The Ontology Alignment Evaluation Initiative (OAEI) could be benefited from UMLS-Meta so that it could be used as the input dataset for a new challenging track within the evaluation campaign. However, in our previous work [3, 4], we showed that UMLS-Meta contains a significant number of logic errors when the rich semantics of the ontology sources is taken into account together with the UMLS-Meta mappings.

2 Method and Results

Our experiments were based on the UMLS-Meta version 2009AA and the corresponding versions of FMA (version 2.0), NCI (version 08.05d) and SNOMED CT (version 20090131), which contain 66,724, 78,989 and 304,802 entities, respectively. After extracting the relevant parts of UMLS, we obtained 3,024 mapping axioms between FMA and NCI, 9,072 between FMA and SNOMED CT and 19,622 between SNOMED CT and NCI. Note that mappings are considered as OWL 2 axioms (see [5, 4]).

When reasoning over each of the source ontologies independently, all their entities were found satisfiable. However, after the respective integrations via UMLS-Meta mappings, we obtained a huge number of unsatisfiable entities, namely 5,015 when integrating FMA and NCI, 16,764 with FMA and SNOMED CT, and 76,025 with SNOMED CT and NCI.

We designed three logic-based principles [3, 4], namely *conservativity principle*, *consistency principle* and *locality principle*, to automatically detect and repair conflictive set of mappings. After the assessment, our automatic methods removed 570 (19%) of the mappings between FMA and NCI, 4,077 (45%) of those between FMA and SNOMED CT and 13,358 (63%) of those between SNOMED CT and NCI. When reasoning with the new revised mapping sets, we found only 2 unsatisfiable entities when integrating FMA and NCI, 44 for FMA and SNOMED CT, and none for SNOMED CT and NCI. These remaining errors were analyzed with our semi-automatic tool ContentMap [5] and they required to repair inherent incompatibilities between the source ontologies (see [4]).

3 Discussion

UMLS-Meta represents a reference of correspondences among ontologies, however, the direct integration of these ontologies with UMLS-Meta mappings leads to a huge number of unintended logical consequences. We propose instead a revised set of UMLS-Meta mappings which could be considered as a silver standard to evaluate ontology matching techniques over FMA, NCI and SNOMED CT. The proposed silver standard and related sources are available for download in `http://krono.act.uji.es/people/Ernesto/umlsassessment`.

The silver standard could be improved by ontology matching tools if they are able to find new valid correspondence which do not lead to logic errors. We also intend to design less aggressive techniques in order to preserve the maximum number of UMLS-Meta mappings. Additionally, ontology sources could be revised before the automatic assessment in order to smooth the incompatibilites.

References

[1] Bodenreider, O.: The unified medical language system (UMLS): integrating biomedical terminology. Nucleic acids research **32**(Database issue) (2004) 267–270

[2] Geller, J., Perl, Y., Halper, M., Cornet, R.: Special issue on auditing of terminologies. Journal of Biomedical Informatics **42**(3) (2009) 407–411

[3] Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Towards a logic-based assessment of the compatibility of UMLS sources. In: Proceedings of the SWAT4LS workshop. Volume 559 of CEUR Workshop Proceedings. (2009)

[4] Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Logic-based assessment of the compatibility of UMLS ontology sources. Accepted for publication in BMC Journal of Biomedical Semantics (2010)

[5] Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: Ontology integration using mappings: Towards getting the right logical consequences. In: Proc. of ESWC. Volume 5554 of LNCS., Springer-Verlag (2009) 173–187

From French EHR to NCI ontology via UMLS

Paolo Besana[1], Marc Cuggia[1], Oussama Zekri[2], Annabel Bourde[1],Anita Burgun[1]

[1] Université de Rennes 1, [2] Centre Régional de Lutte Contre Le Cancer - Eugène Marquis

Abstract. Clinical trials are required for evaluating new therapies and diagnostic techniques. We developed a system based on OWL and SWRL aimed at suggesting the clinical trials to which a patient could be enrolled, evaluating on real clinical trials and patients from the University Hospital of Rennes. This paper presents the method employed to map the French expressions from the patient data to terms in the NCI ontology.

1 Introduction

Clinical trials are fundamental for testing new therapies or diagnostic techniques. Patients are enrolled to a clinical trial if they match its eligibility criteria. We developed a work aimed at suggesting the clinical trials to which a patient could be enrolled [2]. We represent patient data from the Electronic Health Record (EHR) and the eligibility criteria using the NCI Thesaurus [4], an oncology-specific ontology developed by the National Institutes of Health. The project evaluation is based on the clinical trials and patients' data from the Centre Hospitalier Universitaire of Rennes (France) . This paper presents the work done to automatically map the French text of some of the EHR fields to terms from the NCI ontology.

2 Problem description

In order to evaluate the system we extracted 4 real clinical trials active during 2009 in the University Hospital of Rennes, and we selected 486 patients that were assessed for the trials during the same year. Both clinical trials and data were in French. The rules have been converted manually, while the size of the patients data required an automatic processing for at least some of the fields. This paper presents the method employed to automatically map the French expressions of some of the EHR fields to terms from the NCI ontology.

3 Method

A patient's record contains 44 relevant fields. We focussed on the fields containing single expressions. In particular, we chose the field specifying the site of the tumor, very relevant for the recruitment. We extracted the possible different source values $\mathbf{V_{src}} = \left\{v_{src}^1, ..., v_{src}^n\right\}$ of the field . All patients are admitted in urology, and all the cancers are related to urology: the possible values for the field are only 28. Of these values, 16 are composed by more than a single word (for example: 'col de l'uterus' meaning cervix). In particular 5 are specification of a position within an organ. We compared three methods for translation: MESH (Medical Subject Headings, terminology used to index PubMed publications) in French[1], Google translate and Wikipedia. In order to find the NCI concept,

---

[1] http://terminologiecismef.chu-rouen.fr/

Method	Translation found	Correct translation	CUIs found	NCI found	NCI correct
MESH	16 - 57%	16- 57%	13 - 46%	12 - 42%	12 - 42%
Wikipedia	20 - 71%	20 - 71%	21 - 75%	18 - 64%	15 - 53%
Google	28 - 100%	26 - 92%	14 - 50%	9 - 32%	7 - 25%

Table 1. Mapping results for the three different methods on 28 French terms. A translation is found if an expression is returned. The expressions are evaluated manually. The CUIs are concepts in UMLS found by string matching, and the NCI terms are linked to CUIs. The correctness of the NCI terms is checked manually.

we used the UMLS meta-thesaurus (Unified Medical Language System) [3]. In UMLS each concept is identified by a unique key (CUI). A concept is linked to many terms from different terminologies, including NCI. Each term in UMLS is defined by a semantic type (such as Disease or Body Part). Table 1 shows the results for the different methods, discussed more in detail below.

Using Google Translate Google translate was queried for each of the terms in $\mathbf{V_{src}}$. The only error in translation is for a term with homonyms in different domains. The translations were used to query the UMLS by string. To improve relevance, we filtered the results by semantic type. The terms specifying a position in an organ could not be matched directly to defined terms in any of the terminology contained in UMLS. The found CUIs were then filtered to extract the terms from NCI.

Using MESH French Each term in $\mathbf{V_{src}}$ is queried on the MESH French website. The result contains the corresponding English term from the original version of MESH. The English term is then used to query UMLS (filtering by string and by MESH terminology). Of the 16 found translations, 3 were terms belonging to different semantic types and therefore discarded. The resulting CUIs are queried to find the corresponding NCI term.

Using Wikipedia Inspired by the work in [1], each source term is queried in wikipedia French. A possible set of pages is returned. The pages are selected using their wikipedia categories. The filtered pages are used to extract the corresponding English page title. The mapping to NCI follows the same mechanism explained for Google translate. The use of categories to filter unrelated pages improves precision.

References

1. O. Collin B. Gaillard, M. Boualem. Query translation using wikipedia resources for analysis and disambiguation. In *Proceeding of EAMT 2010, Conference of the European Association for Machine Translation*, 2010.
2. P. Besana, M. Cuggia, O. Zekri, A. Bourde, and A. Burgun. Using semantic web technologies for clinical trial recruitment. In *International Semantic Web Conference 2010*, volume 6414. Springer, 2010.
3. O. Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(Database Issue):D267, 2004.
4. N.F. Noy, S. de Coronado, H. Solbrig, G. Fragoso, F.W. Hartel, and M.A. Musen. Representing the NCI Thesaurus in OWL DL: Modeling tools help modeling languages. *Applied ontology*, 3(3):173–190, 2008.

From Mappings to Modules: Using Mappings to Identify Domain-Specific Modules in Large Ontologies

Amir Ghazvinian, Natalya F. Noy, Mark A. Musen

Stanford University, Stanford, CA 94305, US
{amirg,noy,musen}@stanford.edu

Using Mappings to Identify Modules. Ontology modularization is an active area of research in the Semantic Web community [3]. With the emergence and wider use of very large ontologies, in particular in fields such as biomedicine, more and more developers need to extract meaningful modules of these ontologies to use in their applications. Researchers have also noted that many ontology-maintenance tasks would be simplified if we could extract modules from ontologies. These tasks include ontology matching: If we can separate ontologies into modules, we can simplify and improve ontology matching. We study a complementary problem: Can we use existing mappings between ontologies to facilitate modularization?

Methods. Figure 1 illustrates our method: First, we generate mappings from a source ontology to the target ontology that we wish to modularize. Next, we cluster mappings within the target ontology. Finally, we use mapping clusters to identify modules within an ontology.

Validation and Analysis. We validate and analyze our approach by applying our methods to identify modules for NCI Thesaurus [1] and SNOMED-CT [2], two popular and large biomedical ontologies. As domain-specific ontologies for modularization, we used 141 ontologies in BioPortal.[1] Our process extracted 71 modules for NCI Thesaurus and 68 modules for SNOMED-CT. We examined modules and their representative terms in order to understand the types of modules that our algorithm creates and to determine whether or not these modules are likely to be useful in an application setting. Figure 2 shows an example, a module of NCI Thesaurus that is relevant to electrocardiograms (EKG) using the Electrocardiography Ontology. The module consists of 61 classes, representative samples of which are shown in the figure. Of the 61 classes in this module, 41 (67%) are mapping targets.

Discussion and Conclusions. Our approach uses mappings between ontologies in order to extract domain-specific modules from large ontologies based on their mappings to smaller ontologies. In our experiments with NCI Thesaurus and SNOMED-CT, using the ontologies from BioPortal as the sources for mappings, we have identified a number of useful modules. We found that one of the key hurdles that we must overcome, is to find a way to determine how good a particular module is. Indeed, the same problem is true for most modularization approaches [3]: many authors discuss computational properties of their modules, but do not evaluate how useful these modules are to users. In our case, the requirements for extraction are driven by domain coverage of the module rather than by its computational or structural properties. Thus, the problem of evaluating whether the module satisfies the user requirements is similar to the problem of ontology

---

[1] http://bioportal.bioontology.org

evaluation in general: how do we know that an ontology is useful for a specific class of applications? We plan to submit the modules that we have identified to BioPortal to enable the user community to use the modules in their applications, review them and comment on them. However, our initial evidence, which we present in this paper, indicates that our approach can indeed find interesting domain-specific modules.

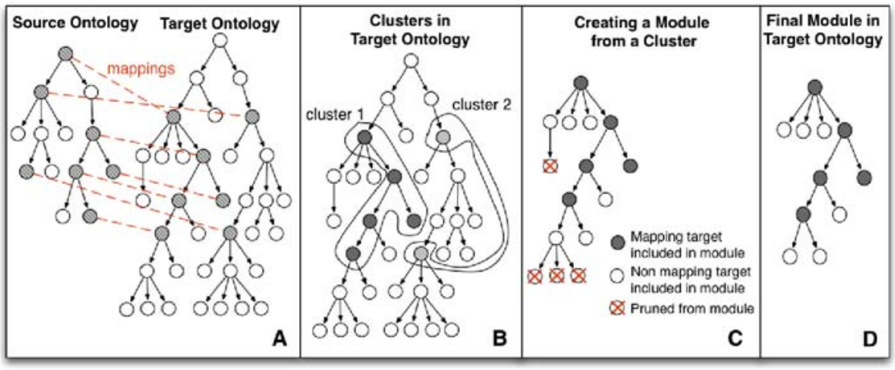


Fig. 1. The process of identifying modules using mappings between ontologies. A: mappings between a source ontology and a modularization target. B: two clusters returned by clustering the mappings. One cluster is light gray in color while the other is dark gray. When determining a module based on these clusters, we discard the light gray cluster since the mapping targets within that cluster are too sparse. C: the process of pruning the ontology subtree for the remaining cluster, which we use to create the module. We begin at each leaf and traverse the tree toward the root, removing all classes that are not mapping targets or direct children of mapping targets. Once we reach such a class, we stop pruning along that branch. D: the final module.

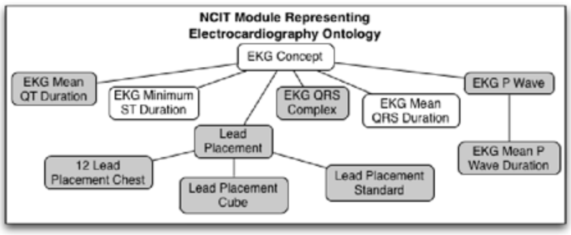


Fig. 2. A portion of the module that we identified within NCI Thesaurus that represents the domain of the Electrocardiography Ontology. The classes in gray represent mapping targets and classes in white represent classes that were not mapping targets, but are included in the module through our algorithm.

References

1. N. Sioutos, S. de Coronado, M. Haber, F. Hartel, W. Shaiu, and L. Wright. NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, 40(1):30–43, 2007.
2. K. Spackman, editor. *SMOMED RT: Systematized Nomenclature of Medicine, Reference Terminology*. College of American Pathologists, Northfield, IL, 2000.
3. H. Stuckenschmidt, C. Parent, and S. Spaccapietra, editors. *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*. Springer-Verlag, Berlin, 2009.

Harnessing the power of folksonomies for formal ontology matching on-the-fly

Theodosia Togia, Fiona McNeill and Alan Bundy

School of Informatics, University of Edinburgh, EH8 9LE, Scotland

Abstract. This paper is a short introduction to our work on building and using folksonomies to facilitate communication between Semantic Web agents with disparate ontological representations. We briefly present the *Semantic Matcher*, a system that measures the semantic proximity between terms in interacting agents' ontologies at run-time, fully automatically and minimally: that is, only for semantic mismatches that impede communication. The system is designed to allow agents to "understand" the meanings of terms to be matched by comparing their folksonomy-based "mental representations".

1 Introduction

The Semantic Matcher is an extension of the *Ontology Repair System* (ORS) [2], a plug-in for a service-requesting agent (*requester*) in the Semantic Web. Terms unknown to the requester which are encountered during interaction with a service-providing agent (*provider*) are mapped to terms in the former's ontology. We assume that the requester, and therefore ORS, has no access to the provider's ontology beyond what is revealed during interation: we are thus concerned with matching not two full ontologies but only individual terms from the provider's ontology to the most relevant terms of the requester's ontology.

We believe that the most serious obstacle for meaning sharing between agents is the lack of symbol grounding in ontologies: ontology terms are unable to refer to the objective world without human interpretation. We argue [3] that this can be dealt with if we allow agents to interprete the meanings of their terms by building a mental representation (sense) [1] for each one of these terms. In our work, senses are simulated by broad folksonomies [4] which annotate physical or abstract resources as opposed to digital resources (e.g. the set of cats in the world vs. a web-page about cats). Folksonomies are created and related to the requester's (formal) ontology. We show that combining ontologies and folksonomies in this way can allow fast and effective matching to be done on-the-fly and provides a way of grounding terms in ontologies to real-world entities.

2 Using Folksonomies for Ontology Matching

The architecture of our matcher is inspired by that of search engines. Broad folksonomies (comparable to "virtual documents" [5] or bags of words) are built for every candidate term (i.e. name of a relation, class or individual) in the requester's ontology with our *sense creation algorithm* (which extracts information from databases such as WordNet and SUMO and manipulates it with techniques such as stemming and stopping[1]). During agent interaction, when ORS diagnoses a semantic mismatch, a sense must be created for the unknown term, which will act as a query to the search engine. This step must be performed on-the-fly without interrupting normal interaction more than necessary. Our search engine then takes the sense representing the unknown term and a list of senses representing the requester's candidate terms as input and returns as output a ranking of the candidate terms.

3 Implementation and Evaluation

The system briefly discussed here has been fully implemented. Evaluation was performed using different versions of the SUMO ontology and its sub-ontologies from the Sigmakee repository[2]. When terms are changed between SUMO versions (e.g. "Corn" becomes "Maize"), we have an objective way of measuring the performance of the matcher because we can safely regard terms and their renamings as synonyms and compare these pairings with our system's prediction. Initial results are encouraging, with 57% of correct matches chosen as the best by the system, 19% as the second-best.

4 Further Work and Conclusions

This paper briefly introduced our work on integrating folksonomies with formal ontologies to perform matching on-the-fly, whenever the need becomes apparent. We believe these ideas could be a major step forward in the problem of ontology matching in an agent communication environment, and in providing symbol grounding for ontology terms. Furthermore, they can provide a framework for the design of matchers which exploit the vast amount of tag data available on the web. Full details of the theory on which this work is based, together with full descriptions of the implementation and evaluation, can be found in [3]. We are currently extending this work and evaluating it more fully.

References

1. G. Frege (1892). On sense and reference. In P. Ludlow, editor, *Readings in the Philosophy of Language*. MIT, 1997.
2. F. McNeill and Alan Bundy. Dynamic, automatic, first-order ontology repair by diagnosis of failed plan execution. *IJSWIS (International Journal on Semantic Web and Information Systems) special issue on Ontology Matching*, 3:1–35, 2007.
3. T. Togia. Automated ontology evolution: Semantic matching. Master's thesis, University of Edinburgh, 2010. Available online at http://dream.inf.ed.ac.uk/projects/dor/ (unpublished).
4. T. Vander Wal. Explaining and showing broad and norrow folksonomies, 2005. available online at http://www.vanderwal.net/random/entrysel.php?blog=1635.
5. W. Hu Y. Qu and G. Cheng. Constructing virtual documents for ontology matching. In *Proceedings of the 15th International WWW Conference*, pages 23–31, 2006.

---

[1] Supplementing this data with folksonomies discovered on the web (e.g. tags that often co-occur with the tag "cat") is also possible though not currently implemented.

[2] http://sigmakee.cvs.sourceforge.net/viewvc/sigmakee/KBs/

Concept abduction for semantic matchmaking in distributed and modular ontologies

Viet-Hoang VU, Nhan LE-THANH

KEWI - Laboratory I3S - CNRS
Nice Sophia Antipolis University, France.

Abstract. Recently, attempts have been done to formalize Semantic Matchmaking, the process of finding potential matches between demands and supplies based on their logical relations with reference to a common ontology, in Description Logics (DLs). We extend the formalization to Packages-based Description Logics (P-DLs), modular extensions of DLs, in order to perform the matchmaking operation in contexts where descriptions of demands and supplies are specified in different terminologies.

Keywords: semantic matchmaking, package-based description logics, distributed ontology, abductive reasoning, tableaux-based methods.

1 Introduction

Recently, several attempts have been done to formalize Semantic Matchmaking, the process of finding potential matches based on logical relations with reference to a *common ontology*, in Description Logics (DLs)[4]. Demands and supplies are thus represented as *concepts* in order to use different inference mechanisms to evaluate possible matches between them according to the semantic relationships. We extend this formalization to the context of Packages-based Description Logics (P-DLs)[2] to allow descriptions of demands and supplies can be specified in *different, distributed* but *interconnected ontologies.* For performing the semantic matchmaking in such settings, we develop a distributed reasoning algorithm based on tableau calculus to compute *concept abduction*, an abductive reasoning service developed specifically for this kind of operation[4], in P-DLs.

2 Distributed abductive reasoning in Package-based Description Logics

Description Logics (DLs)[1] are a family of logic-based languages for representing and reasoning about the knowledge of a domain. Package-based Description Logics (P-DLs) are extensions of DLs to represent distributed and modular ontologies. In P-DLs, a knowledge base (KB) is thus considered as a collection of components called *packages.* In each package, along with its *local terms*, the usage of *foreign terms* imported from other packages are permitted, allowing local knowledge of packages can be reused elsewhere.

Concept Abduction is a novel non-monotonic inference task proposed for description logics to evaluate potential matches between demands and supplies. Given two concepts C, D and a TBox $\mathcal{T}$ such that $C \sqcap D \not\sqsubseteq_{\mathcal{T}} \bot$, this reasoning service allows to find a concept H (hypothesis) such that $C \sqcap H \not\sqsubseteq_{\mathcal{T}} \bot$ and $C \sqcap H \sqsubseteq_{\mathcal{T}} D$.

Extending Concept Abduction to the distributed context of P-DLs, in place of a single TBox $\mathcal{T}$, we have a set of packages $\Sigma = \{P_i\}$. Let $P_w \in \Sigma$ be some *witness package*, the computing of H now need to be done with respect to P_w . To do that, we devise a distributed algorithm which is based on the federated reasoning technique developed for P-DLs[2] and the uniform tableaux-based method[3]. The procedure consists of two stages:

1. At first, we try to build multiple, federated local tableaux for $C \sqsubseteq_w D$[1]. If all such tableaux contain obvious contradiction, then either C is unsatisfiable or the subsumption holds w.r.t Σ as witnessed by P_w and thus no abduction is needed.
2. On the contrary, for all tableaux which are consistent, we compute concept expressions that, when added to the tableaux, will eventually generate some contradiction. If these contradictions imply in fact the subsumption between two concepts C and D, they will be parts of the finding hypothesis H.

3 Conclusions

We have developed a distributed tableaux-based algorithm for solving the concept abduction problem in the distributed and modular context of Package-based Description Logics. This allows to perform the Semantic Matchmaking in situations where demands and supplies are specified with reference to different ontologies. In the future, we would like to optimize this method to achieve better performance.

References

1. Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.
2. Jie Bao, Doina Caragea, and Vasant Honavar. A tableau-based federated reasoning algorithm for modular ontologies. In *In IEEE/WIC/ACM International Conference on Web Intelligence*, pages 404–410. IEEE Press, 2006.
3. S. Colucci, T. Di Noia, E. Di Sciascio, F. M. Donini, and M. Mongiello. A uniform tableaux-based approach to concept abduction and contraction in aln. In *Contraction in ALN . In Proc. of DL-04 (2004), CEUR Workshop Proceedings*, page 104, 2004.
4. Tommaso Di Noia, Eugenio Di Sciascio, and Francesco M. Donini. Semantic matchmaking as non-monotonic reasoning: A description logic approach. *Journal of Artificial Intelligence Research*, 29:307, 2007.

---

[1] '$\sqsubseteq_w$' is used to denote that the inference is performed from the local point of view of P_w.

LingNet: Networking Linguistic and Terminological Ontologies

Wim Peters
Department of Computer Science
University of Sheffield, UK
w.peters@dcs.shef.ac.uk

Abstract. Linguistic description and linguistically-based re-engineering techniques depend on standardized models. LingNet aims at the alignment of multiple standard description models from the terminological, linguistic and localization fields.

1 Motivation

Linguistic and terminological standards are in daily use for the purpose of linguistic and terminological resource creation (term banks, dictionaries, translation memories etc.). Since there is a variety of standard models available. Examples of terminological and linguistic models are the ISO standard initiatives ISO16620 and the Lexical Markup Framework (LMF) [1]. Furthermore, the purpose of e.g. LexInfo [2] and the Linguistic Information Repository (LIR) [3] is to associate multilingual linguistic knowledge with conceptual ontology elements. In the translation memory area standards such as TMX[1] (Translation Memory eXchange) and XLIFF[2] (XML Localization Interchange File Format) are widely used.

In order to enable interdisciplinary re-use and complementarity it is necessary to establish interoperability between their vocabularies in a principled way. **LingNet**[3] is a model for mapping linguistic and terminological (standard) information in a distributed fashion. The LingNet model adopts a number of modelling decisions from the literature: a knowledge-based, formalism-independent metamodel for capturing semantic alignments between ontologies for linguistic/terminological description [4], references to external mapping patterns for structural mappings [5], and a the integration of a lexicalization relation for linking linguistic/terminological resource to ontological concepts [6].

The novelty of the LingNet model lies in its combination of selected mapping methods and its application to the linguistic/terminological domain. The basic structure of LingNet is illustrated in figure 1 below.

---

[1] http://www.lisa.org/tmx/

[2] http://docs.oasis-open.org/xliff/v1.2/os/xliff-core.pdf

[3] first version: http://www.gate.ac.uk/ns/ontologies/LingNet/LingNet-v0.1.owl.

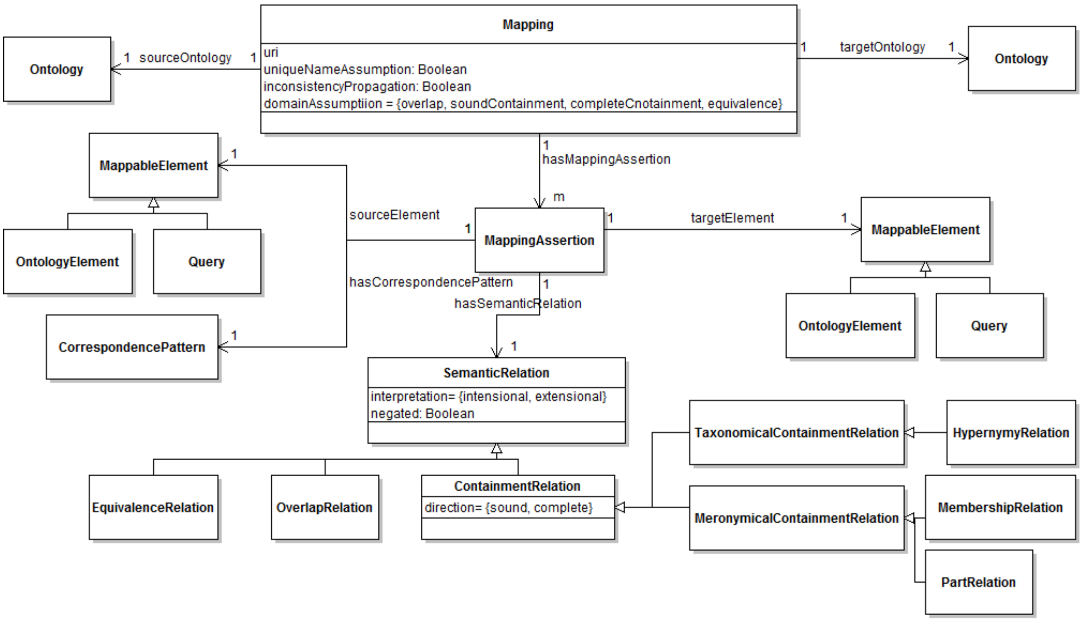


Fig. 1. The LingNet Model

References

1. Francopoulo, G., Monte George, Nicoletta Calzolari, Monica Monachini, Nuria Bel, Mandy Pet, Claudia Soria: LMF for multilingual, specialized lexicons. In: LREC, Genova, Italy (2006)

2. Buitelaar, P., Cimiano, P., Haase, P. and Sintek, M.: Towards Linguistically Grounded Ontologies, in: The Semantic Web: Research and Applications, 6th European Semantic Web Conference, proceedings of ESWC 2009, Heraklion, Crete, Greece, Springer Berlin / Heidelberg, Volume 5554 (2009)

3. Peters, W. Montiel-Ponsoda, E. Aguado de Cea, G.: Localizing Ontologies in OWL. In: Proceedings of the ISWC07 OntoLex workshop, Busan, Korea (2007)

4. Brockmans, S., Haase, P., Stuckenschmidt, H.: Formalism-Independent Specification of Ontology Mappings - A Metamodeling Approach, In: Robert Meersman, R., Tari, Z. et al. (eds), OTM 2006 Conferences, Springer Verlag, Montpellier, France (2006)

5 Scharffe, F. Euzenat, J. and Fensel, D.: Towards design patterns for ontology alignment. In R.L. Wainwright and H. Haddad (eds.): Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil, March 2008: 2321-2325 (2008)

6 Picca, D., Gangemi, A., and Gliozzo, A.: LMM: an OWL Metamodel to Represent Heterogeneous Lexical Knowledge. In Proc. of the International Conference on Language Resources and Evaluation (LREC), Marrakech, Morocco (2008)

Aggregation of Similarity Measures in Ontology Matching

Lihua Zhao[1] and Ryutaro Ichise[2]

[1] National Institute of Informatics, Tokyo, Japan, `lihua@nii.ac.jp`
[2] National Institute of Informatics, Tokyo, Japan, `ichise@nii.ac.jp`

Abstract. This paper presents an aggregation approach of similarity measures for ontology matching called n-Harmony. The n-Harmony measure identifies top-n highest values in each similarity matrix to assign a weight to the corresponding similarity measure for aggregation. We can also exclude noisy similarity measures that have a low weight and the n-Harmony outperforms previous methods in our experimental tests.

1 Introduction

Ontology matching is a promising research field that discovers similarities between two ontologies and is widely used in applications such as semantic web, biomedical informatics and software engineering. Most of current ontology matching systems combine different similarity measures. For instance, the authors in [4] applied the Ordered Weighted Average(OWA) to combine similarity measures and Ichise[3] proposed a machine learning approach to aggregate 40 similarity measures.

Harmony[5] measure is a state-of-the-art adaptive aggregation method that assigns a higher weight to reliable and important similarity measure and a lower weight to those fail to map similar ontologies. The harmony weight for a similarity measure is calculated according to the number of the highest values in the corresponding similarity matrix. However, the harmony measure has drawbacks when there exist other similarity measures that are as important as the ones with the highest similarity value. Hence, we extended the harmony measure by considering top-n values in each row and column of similarity matrices and we call this method as n-Harmony measure. The top-n is calculated according to the number of concepts in two ontologies. Our extended n-Harmony considers more values in similarity matrices and only aggregates similarity measures that have a high harmony weight.

2 n-Harmony Measure

We applied 13 different similarity measures for aggregation which include 4 string-based, 1 structure-based and 8 WordNet-based similarity measures[2]. The final aggregated similarity matrix is $\frac{\sum_k (nH_k \times SMatrix_k(O_s,O_t))}{|SMatrix|}$, where nH_k is n-Harmony weight and SMatrix is the similarity matrix of each similarity measure between ontology O_s and O_t. Before combining the similarity matrices, we remove min(L-1, nH × L) lowest values in each row and column of similarity matrix, where L is the minimum number of concepts in two ontologies and nH represents harmony weight of corresponding similarity matrix. Furthermore, only those similarity matrices with a high harmony weight are aggregated for the final similarity matrix. The final decision of whether a ontology pair is matching or not depends on the final similarity matrix and manually tuned threshold.

Directory data sets[3] and Benchmark data sets[4] from OAEI[5] are tested with our system. The n-Harmony measure returns best result on Directory data sets when the threshold is 0.45 with 0.86 recall and 0.70 F-measure while the original harmony measure returns 0.81 recall and 0.68 F-measure. This result is also better than the results of best systems in OM2009[1], such as ASMOV which reaches 0.65 recall and 0.63 F-measure on the Directory data sets. On the Benchmark data sets, n-Harmony performs the same as the original harmony measure or returns slightly better recalls and F-measures than harmony. Comparing with the ASMOV, n-Harmony performs almost the same on data sets #101-104 and #221-247 and returns higher precisions on #302-304, but slightly lower recalls.

3 Conclusions and Future Work

Experimental results show that our n-Harmony outperforms original harmony measure on most of the Directory and Benchmark data sets and also comparable with the best systems attended in OM2009. However there are still rooms to improve our n-Harmony measure by exploring advanced structure-based similarity measures and by investigating automatic threshold selection method rather than manually tuning the threshold to find out the best performance.

References

1. Euzenat, J., et al.: Results of the ontology alignment evaluation initiative 2009. In: Proceedings of the 4th International Workshop on Ontology Matching. pp. 82–135 (2009)
2. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (2007)
3. Ichise, R.: Machine learning approach for ontology matching using multiple concept similarity measures. In: Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science. pp. 340–346 (2008)
4. Ji, Q., Haase, P., Qi, G.: Combination of similarity measures in ontology matching using the owa operator. In: Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Sytems. pp. 243–250 (2008)
5. Mao, M., Peng, Y., Spring, M.: An adaptive ontology mapping approach with neural network based constraint satisfaction. In: Web Semantics: Science, Services and Agents on the World Wide Web. pp. 14–25. No. 8 (2010)

---

[3] http://oaei.ontologymatching.org/2009/directory/

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

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

Using Concept and Structure Similarities for Ontology Integration

Xiulei Liu[1][2], Payam Barnaghi[1], Klaus Moessner[1], and Jianxin Liao[2]
[1]Centre for Communication Systems Research University of Surrey
Guildford, Surrey, GU2 7XH, United Kingdom
[2]State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications Beijing 100876, P.R. of China
`{xiulei.liu,p.barnaghi,k.moessner}@surrey.ac.uk`
`liaojianxin@ebupt.com`

Abstract. We propose a method to align different ontologies in similar domains and then define correspondence between concepts in two different ontologies using the SKOS model.

Introduction. Recently ontologies are created to provide knowledge representation. They use common representation languages such as OWL, but there are many heterogeneous ontologies [1–3]. In this paper we first propose a lexical and structural analysis and compute the concept similarity as a combination of attributes, second use the SKOS model to define correspondence between concepts[4].

Ontology Alignment Framework. To perform the matching between concepts in different ontologies, we focus both on syntactical and text in entity descriptions and also their semantic structure in the ontology representations. This process, illustrated in the block diagram shown in Figure 1, is divided into two main sub-tasks: Alignment and SKOS translation. The inputs are two ontologies and result of the process is an SKOS-based ontology that contains automatically defined associations. The alignment task analyses lexical and structural attributes of ontologies to automatically produce associations between concepts. The relation is defined: $\mathcal{R}(\mathcal{A},\mathcal{B}) =< \mathcal{A},\mathcal{B},\mathcal{R}elation,\mathcal{S}(A,B) >$ where $\mathcal{A}$ and $\mathcal{B}$ are ontology concepts, $\mathcal{R}elation$ describe semantic relations between these concepts which have five types: equal beIncluded, include, disjoint, related, and $\mathcal{S}(A,B)$ is similarity measure for two concepts based on their structure and lexical analysis.

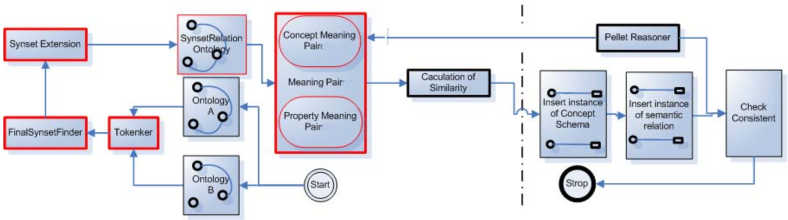


Fig. 1. The ontology alignment process

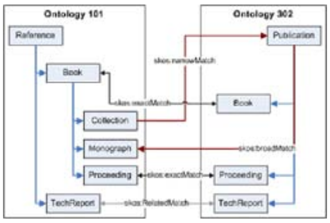


Fig. 2. Snapshot of the specified properties in the integrated ontology

Defining SKOS-based Associations. After identifying possible relations between concepts, they are imported based on the SKOS model. This will provide an interconnection between two ontologies based on standard set of properties defined in the SKOS model. The SKOS mapping properties include *skos:closeMatch*, *skos:exactMatch*, *skos:broadMatch*, *skos:narrowMatch* and *skos:relatedMatch*. The properties maintain a mapping between SKOS concepts adapted from schemes. The relations in concept pairs defined in the previous section are based on *synset* relations in WordNet. They are obtained according to accessing the extended *synset* collection for each representative word that describes entities and calculating structural similarity We will map between *synset* and SKOS relations. By applying these mappings, the final product of the ontology integration process will include assertion axioms in which the related concepts from different ontologies are linked to each other based on SKOS relations. The integrated ontology will be a collection of concepts and properties from both ontologies and will also include the SKOS association properties. Figure 2 illustrates a part of the SKOS relations and concept alignment between two ontologies from the dataset (a complete set of our evaluation results using OAEI2008 dataset can be accessed from: `http://tinyurl.com/38veolh`).

Acknowledgement. The work is partially supported by the *m:Ciudad* project funded by the European 7th Framework Programme, contract number: 215007. Xiulei Liu's and Jianxin Liao's research is supported by Chinese National Science Fund for Distinguished Young Scholars (No. 60525110) and Chinese National 973 Program (No. 2007CB307100, 2007CB307103).

References

1. T. L. Bach, J. D. Bo, and R. Lara, "Knowledgeweb consortium, state of the art on ontology (d2.2.3)," 2004.
2. P.Shvaiko and J.Euzenat, "A survey of schema-based matching approaches," in J. Data Semantics IV, 2005, pp. 146–171.
3. Y. Kalfoglou and M. Schorlemmer, "Ontology mapping: The state of the art," 2003.
4. A. Miles and S. Bechhofer, "(editors) skos simple knowledge organization system reference, w3c recommendation 18 august 2009." [Online]. Available: http://www.w3.org/TR/skos-reference/

Flexible Bootstrapping-Based Ontology Alignment

Prateek Jain, Pascal Hitzler, Amit P. Sheth

Kno.e.sis Center, Wright State University, Dayton, OH

Abstract. BLOOMS (Jain et al, ISWC2010, to appear) is an ontology alignment system which, in its core, utilizes the Wikipedia category hierarchy for establishing alignments. In this paper, we present a Plug-and-Play extension to BLOOMS, which allows to flexibly replace or complement the use of Wikipedia by other online or offline resources, including domain-specific ontologies or taxonomies. By making use of automated translation services and of Wikipedia in languages other than English, it makes it possible to apply BLOOMS to alignment tasks where the input ontologies are written in different languages.

1 Introduction

In this work, we present the extension of BLOOMS called Plug-n-Play BLOOMS (PnP BLOOMS). PnP BLOOMS allows users to plug in a datasource of choice to assist in the task of ontology matching while utilizing the core BLOOMS approach. Thus, PnP BLOOMS can be customized by plugging in other information sources best suited to the needs of the domain and the user. At the same time it has been developed in a modular fashion to allow for de-coupling of the main matching technique with the auxiliary source. The system is available from http://wiki.knoesis.org/index.php/BLOOMS.

The paper is organized as follows. In Section 2, we present our proposed solution, Section 3 contains a minimalistic evaluation which shows the feasibility of our approach and finally we conclude in Section 4.

2 Our Solution

We present the BLOOMS approach following [1], modified as appropriate to cater for our plug-n-play extension. BLOOMS constructs a forest (i.e., a set of trees) T_C (which we call the *BLOOMS forest* for C) for each matching candidate class name C – in the original approach, this roughly corresponds to a selection of supercategories of the class name in the sense of the Wikipedia clas shierarchy. Comparison of the forests T_C and T_B for matching candidate classes C and B then yields a decision whether or not (and with which of the candidate relations) C and B should be aligned.

We have implemented an extension of the BLOOMS framework from [1] to support 'plug-n-play' methodology. Plug-n-Play methodology provides users the flexibility to customize the framework to use the best auxiliary datasource based on their needs and application scenario. The framework allows users to plug-in the auxiliary data source which is best suited to the target domain. This is in part possible due to the nature of the BLOOMS approach, which relies on the comparison of nodes in the categorization tree.

Table 1. Comparison of BLOOMS and PnP BLOOMS on French ontologies of the Benchmark track of OAEI 2009.

Ontology	f-measure BLOOMS	f-measure PnP BLOOMS
206	0.53	0.76
207	0.58	0.77
210	0.50	0.72
Avg.	0.54	0.75

3 Evaluation

In order to evaluate our plug-n-play approach for ontology matching, we utilized the French language ontologies from the Benchmark track of OAEI 2010 initiative.[1] We utilized these ontologies since It allows us to compare the performance of PnP BLOOMS with the old version of BLOOMS. Further, it allows us to check the plug-n-play version of BLOOMS with respect to ease of use and functionality. Table 1 illustrates the comparative performance of PnP BLOOMS with BLOOMS. The table clearly illustrates the advantage obtained by using the PnP approach. The effectiveness of the BLOOMS PnP approach is demonstrated by an increment of 42% over the previous version of BLOOMS.

4 Conclusion and Future Work

In this paper we have presented an extension of our recent work on BLOOMS [1] which allows users to plug-in auxiliary sources of their choice as oracles in the task of ontology matching. It differs from the previous version of BLOOMS which was restricted to using English language Wikipedia to help in this task. The effectiveness of the approach is demonstrated by the increment of 42% over the previous version of BLOOMS. The flexible plug-n-play based approach allows users to customize the system based on their needs and the domain and language of the ontologies. Using PnP BLOOMS, we are also planning on investigating the use of 'cocktails' of auxiliary data sources for the task of ontology matching.

Acknowledgement This work is funded primarily by NSF Award:IIS-0842129, titled "III-SGER: Spatio-Temporal-Thematic Queries of Semantic Web Data: a Study of Expressivity and Efficiency". Pascal Hitzler acknowledges support by the Wright State University Research Council.

References

1. Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology Alignment for Linked Open Data. In: Proceedings of the 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010 (To Appear), Springer-Verlag (2010) 402–417 Available from http://knoesis.org/library/resource.php?id=844.

---

[1] http://oaei.ontologymatching.org/2009/

Semantic Matching of Ontologies

Christoph Quix, Marko Pascan, Pratanu Roy, David Kensche

Informatik 5 (Information Systems), RWTH Aachen University, Germany
`http://www.dbis.rwth-aachen.de`

Introduction The discovery of semantic relationships such as subsumption and disjointness is still a challenge in ontology matching [6]. Existing methods use logical reasoning over computed equivalence relationships or machine learning based on lexical and structural features of ontology elements [7, 3]. While these methods deliver good results for some cases, they are limited to the information contained in the input ontologies to be matched. Therefore, background knowledge in form of an additional ontology may be useful to detect semantic relationships. In existing approaches, the identification of an appropriate ontology as background knowledge is often a task left for the user. We present two enhanced approaches for identifying semantic relationships. The first one is based on background knowledge; in contrast to other approaches, it is able to identify a background ontology automatically. The second approach builds on existing machine learning methods for identifying semantic relationships. First evaluation results for these methods and combined approaches show that the integration of these methods is reasonable as more semantic relationships are identified.

Semantic Matching using Background Knowledge It has been shown in previous works that using an ontology as background knowledge can improve the match result [1]. The selection of the background ontology is obviously an important step in such an approach. While earlier works either relied on the user to provide such an ontology [1], or used very general upper ontologies (e.g. SUMO-OWL, [5]), our approach is able to select the background ontology automatically. The idea is illustrated in fig. 1 and 2. For the input ontologies S and T, we generate keyword queries for a web search engine (e.g., Google or Swoogle), and for our local ontology repository. The external search engine is only used if the local repository does not contain an appropriate ontology. When a background ontology O is found it can be used for matching. In addition to the direct alignment A_{dir}, two alignments $A_{O,S}$ and $A_{O,T}$, between the input ontologies and the background ontology, are computed. Then, for each pair of correspondences from $A_{O,S}$ and $A_{O,T}$, existence of a relationship (i.e., equivalence, subsumption) between the model elements from O is determined. If that is the case, a new correspondence between the concepts from S and T can be inferred. All the correspondences found in this way are called *semantic matches* (A_{sem}). Eventually, the final result is created by building the union of A_{dir} and A_{sem}.

Using Machine Learning for Semantic Matching Our second, complementary method uses machine learning to identify semantic relationships. We implemented an approach similar to the method presented in [7]. The computation of subsumption relationships is considered as a binary classification task, i.e., a concept pair is classified into two possible classes: subsumption and not-subsumption relationship.

Because ontologies are usually hierarchical structures, the subsumption relationships found in input ontologies can be used as training examples, making the process of classifier training independent of alignment computation. Each training example is a

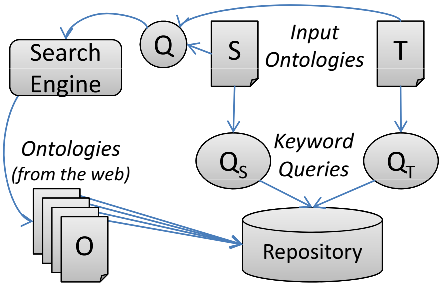


Fig. 1. Selection of the background ontology

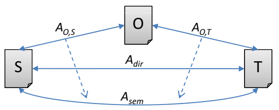


Fig. 2. Matching using a background ontology

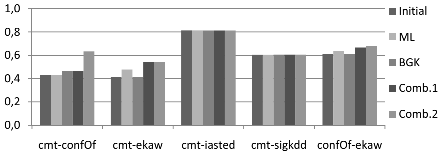


Fig. 3. Recall for OAEI conference track

concept pair (c_i, c_j), where $c_i \sqsubseteq c_j$ and both concepts belong to a single ontology, i.e., the source or target ontology of a matching task. We use distinct words, extracted from both source and target ontologies, as features for the machine learning method. In order to represent the concept pairs in the feature space, each concept c_i of a concept pair is described by a set of feature space words that can be found in its neighborhood, constituting the concept's *context document* D_{c_i}. The notion of the concept's neighborhood can be defined in various ways. In our implementation D_{c_i} is created from words found in: name, label, comment, instances, data and object properties of c_i, direct sub or super concepts of c_i, concepts in union or intersection definitions of c_i and equivalent concepts of c_i. The context documents of concepts are translated to feature vectors which are used as input for the machine learning method (currently, C4.5 decision tree). We use an optimization to avoid the classification of all concept pairs.

Conclusion The presented approaches have been integrated into our generic matching system *GeRoMeSuite* [4]. To evaluate our approach, we used *semantic* precision and recall as measures [2] and data sets from the oriented track of OAEI 2009. We created also two combined match configurations using both approaches. The results in fig. 3 show very good results for the combined approaches.

Acknowledgments: This work is supported by the Umbrella Cooperation Programme (`http://www.umbrella-coop.org/`).

References

1. Z. Aleksovski, M. Klein, W. ten Kate, F. van Harmelen. Matching Unstructured Vocabularies Using a Background Ontology. *Proc. EKAW*. 2006.
2. J. Euzenat. Semantic precision and recall for ontology alignment evaluation. *Proc. IJCAI*, pp. 348–353. 2007.
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, X. Li, Y. Li. *GeRoMeSuite*: A System for Holistic Generic Model Management. *Proc. VLDB*, pp. 1322–1325. 2007.
5. V. Mascardi, A. Locoro, P. Rosso. Automatic Ontology Matching via Upper Ontologies: A Systematic Evaluation. *IEEE Trans. on Knowl. & Data Eng.*, **22**:609–623, 2010.
6. P. Shvaiko, J. Euzenat. Ten Challenges for Ontology Matching. *Proc. ODBASE*, pp. 1164–1182. 2008.
7. V. Spiliopoulos, G. A. Vouros, V. Karkaletsis. On the discovery of subsumption relations for the alignment of ontologies. *Web Semantics*, **8**(1):69–88, 2010.

Ontology Mapping Neural Network: An Approach to Learning and Inferring Correspondences among Ontologies

Yefei Peng[1]⋆, Paul Munro[1], and Ming Mao[2]

[1] University of Pittsburgh, Pittsburgh PA 15206, USA,
`yep3@pitt.edu, pmunro@pitt.edu`,
[2] SAP Labs, Palo Alto CA 94304, USA,
`ming.mao@sap.com`,

The Ontology Mapping Neural Network (OMNN) extends the ability of Identical Elements Neural Network(IENN) and its variants' [4, 1–3] to represent and map complex relationships. The network can learn high-level features common to different tasks, and use them to infer correspondence between the tasks. The learning dynamics of simultaneous (interlaced) training of similar tasks interact at the shared connections of the networks. The output of one network in response to a stimulus to another network can be interpreted as an analogical mapping. In a similar fashion, the networks can be explicitly trained to map specific items in one domain to specific items in another domain. A more detailed version is published on the main conference [5].

The network architecture is shown in Figure 1. A_{in} and B_{in} are input subvectors for nodes from ontology A and ontology B respectively. They share one representation layer AB_r. RA_{in} represents relationships from graph A; RB_{in} represents relationships from graph B. They share one representation layer R_r.

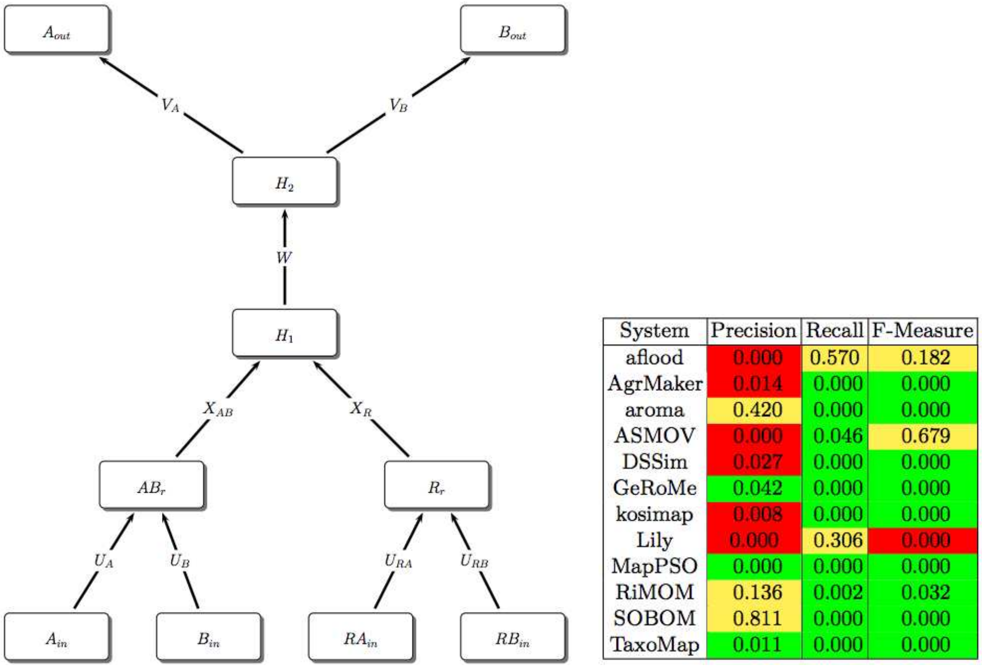
In this network, each to-be-mapped node in graph is represented by a single active unit in input layers (A_{in}, B_{in}) and output layers (A_{out}, B_{out}). For relationships representation in input layer (RA_{in}, RB_{in}), each relationship is represented by a single active unit. The network shown in Figure 1 has multiple sub networks shown in the following list.

1. Net_{AAA} : $\{A_{in}\text{-}AB_r\text{-}X_{AB};\ RA_{in}\text{-}R_{RA}\text{-}X_R\ \}\text{-}H_1\text{-}W\text{-}H_2\text{-}V_A\text{-}A_{out}$;
2. Net_{BBB} : $\{B_{in}\text{-}AB_r\text{-}X_{AB};\ RB_{in}\text{-}R_{RB}\text{-}X_R\ \}\text{-}H_1\text{-}W\text{-}H_2\text{-}V_B\text{-}B_{out}$;
3. Net_{AAB} : $\{A_{in}\text{-}AB_r\text{-}X_{AB};\ RA_{in}\text{-}R_{RA}\text{-}X_R\ \}\text{-}H_1\text{-}W\text{-}H_2\text{-}V_B\text{-}B_{out}$;
4. Net_{BBA} : $\{B_{in}\text{-}AB_r\text{-}X_{AB};\ RB_{in}\text{-}R_{RB}\text{-}X_R\ \}\text{-}H_1\text{-}W\text{-}H_2\text{-}V_A\text{-}A_{out}$;

Selected OAEI [3] benchmark tests are used to evaluate OMNN approach. Wilcox test is performed to compare OMNN with the other 12 systems participated in OAEI 2009 on precision, recall and f-measure. The result is shown in Figure 1. Green means OMNN is significantly better than the system; Red means the system is significantly better than OMNN. Yellow means no significant difference. Significance is defined as $p - value < 0.05$. It shows that OMNN

⋆ The author is working at Google now. Email: `yefeip@google.com`

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



System	Precision	Recall	F-Measure
aflood	0.000	0.570	0.182
AgrMaker	0.014	0.000	0.000
aroma	0.420	0.000	0.000
ASMOV	0.000	0.046	0.679
DSSim	0.027	0.000	0.000
GeRoMe	0.042	0.000	0.000
kosimap	0.008	0.000	0.000
Lily	0.000	0.306	0.000
MapPSO	0.000	0.000	0.000
RiMOM	0.136	0.002	0.032
SOBOM	0.811	0.000	0.000
TaxoMap	0.011	0.000	0.000

Fig. 1. Proposed network architecture and Results

has better F-measure than 9 of the 12 systems, OMNN's recall is significantly better than 10 of the systems. It should be noted that p-value< 0.05 means there is significant difference between two systems compared, then detailed data is visited to reveal which is one is better than the other.

References

1. Munro, P.: Shared network resources and shared task properties. In: Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society (1996)
2. Munro, P., Bao, J.: A connectionist implementation of identical elements. In: Proceedings of the Twenty Seventh Ann. Conf. Cognitive Science Society Proceedings. Lawerence Erlbaum: Mahwah NJ (2005)
3. Munro, P.W.: Learning structurally analogous tasks. In: Kurková, V., Neruda, R., Koutník, J. (eds.) Artificial Neural Networks - ICANN 2008, 18th International Conference. Lecture Notes in Computer Science, vol. 5164, pp. 406–412. Springer: Berlin/Heidelberg (2008)
4. Munro, P., Peng, Y.: Analogical learning and inference in overlapping networks. In: Kokinov, B., Holyoak, K., Gentner, D. (eds.) New Frontiers in Analogy Research. New Bulgarian University Press (2009)
5. Peng, Y., Munro, P., Mao, M.: Ontology mapping neural network: An approach to learning and inferring correspondences among ontologies. In: Proceedings of the 9th International Semantic Web Conference (2010)

Towards Tailored Domain Ontologies

Cheikh Niang[1,2], Béatrice Bouchou[1], and Moussa Lo[2]

[1] Université François Rabelais Tours - Laboratoire d'Informatique
[2] Université Gaston Berger de Saint-Louis - LANI

Introduction. The goal of domain ontology is to provide a common conceptual vocabulary to members of a virtual community of users who need to share their information in a particular domain (such as medical, tourism, banking, agricultural). The identification and definition of concepts that describe the domain knowledge requires a certain consensus. Generally, each member or subcommunity holds some knowledge, he has its own view on the domain, and he describes it with his own vocabulary. Thus, to reach a consensus allowing to reflect a common view of the domain can be a difficult task and even more harder if members are geographically dispersed. One way very widely used is to start from pre-existent elements in the domain: text corpus, taxonomies, ontology fragments, and to exploit them as a basis for gradually defining the domain ontology [2][7].

In this short paper, we present an approach using Ontology Matching techniques [1][5][6][3] for building a tailored domain ontology, starting from a general domain taxonomy and several pieces of knowledge given by different partners.

Our strategy is to design a mediator, firstly to reach an agreement with each partner on their knowledge fragments that will be part to the shared domain ontology, and secondly to conciliate these various fragments by linking and structuring the concepts that compose them. As a mediator ontology, in our case study we use a public taxonomy that exists for describing subject fields in agriculture, forestry, fisheries, food and related domains (e.g. environment), called AGROVOC[3]. The resulting domain ontology combines the following two features: (*i*) it is the portion of the general taxonomy that is relevant to the considered application domain as seen by each partner, (*ii*) it is completed and tailored by relations and properties coming from partner's data. Fig. 1 shows an example of a domain ontology *DO* built starting from two local ontologies LO_1 and LO_2. *DO*'s concepts prefixed with *ag* are from AGROVOC. One can see that in *DO Plan_products* and *Varieties* are related and also that they are related to attributes *price* and *surface*, which is not the case in AGROVOC.

Reaching an agreement with a partner. This is the first step of our general approach. Each partner's fragment knowledge is represented by a **Local Ontology**, denoted by *LO*. The agreement between the mediator and the partner is concluded based on a matching between *LO* and the mediator ontology *MO*. It is consented by the partner that each concept of *LO* which can be associated with a concept of *MO*, called its anchor, will be a concept of the tailored domain

---

[3] http ://www4.fao.org/agrovoc/

[4] This work is supported by ANR-08-DEFIS-04

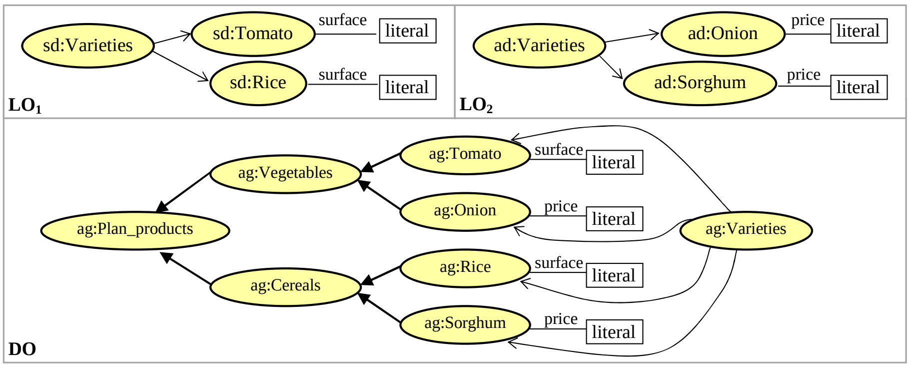


Fig. 1. Domain Ontology built starting from LO_1 and LO_2.

ontology DO. This agreement is also an ontology composed by the anchored concepts of LO with their anchor, as well as the local relationships between them.

Conciliation. Once the mediator has found an agreement with each partner on the concepts which must be part to the domain ontology, it applies a conciliation phase at the end of which the domain ontology is built. This is an incrementaly phase, the local ontologies are conciliated by integrating their agreement into the domain ontology DO, one after another. To achieve efficiently this phase, (i) the mediator ontology is partitioned into blocks, according to Falcon-AO method [4] and (ii) conflict resolution strategies are applied. Each block is a sub-ontology of MO containing semantically close concepts. Our algorithm relies on this classification in order to find links that exist between the concepts already present in the domain ontology and those of the new local ontology to conciliate.

References

1. Zharko Aleksovski, Warner ten Kate, and Frank van Harmelen. Exploiting the structure of background knowledge used in ontology matching. In *Ontology Matching*, 2006.
2. Loris Bozzato, Mauro Ferrari, and Alberto Trombetta. Building a domain ontology from glossaries: A general methodology. In *SWAP*, 2008.
3. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer, 2007.
4. Wei Hu, Yuanyuan Zhao, and Yuzhong Qu. Partition-based block matching of large class hierarchies. In *ASWC*, pages 72–83, 2006.
5. Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: The state of the art. *The Knowledge Engineering Review*, 18:2003, 2003.
6. Pavel Shvaiko and Jerome Euzenat. Ten challenges for ontology matching. In *Proceedings of The 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, 2008.
7. Jie Yang, Lei Wang, Song Zhang, Xin Sui, Ning Zhang 0003, and Zhuoqun Xu. Building domain ontology based on web data and generic ontology. In *Web Intelligence*, pages 686–689, 2004.

Crowd sourcing through social gaming for community driven ontology engineering, results and observations

Alloy Martin Chua, Roland Christian Chua, Arthur Vincent Dychiching, Tinmon Ang, Jose Lloyd Espiritu, Nathalie Rose Lim, Danny C. Cheng

College of Computer Studies, De La Salle University Manila Philippines
{alchua01, arvolution7000, avd_arthur, angtinmon}@yahoo.com,
espiritu.lloyd@gmail.com, nats.lim@delasalle.ph, danny.cheng@dlsu.edu.ph

Abstract. In developing ontology, expert driven approaches lack the scalability to accommodate the vast amount of data on the web. As such, the community is being tapped to build ontologies to cope with highly dynamic data sources. Common problems (like difficulty of the task, quality of output, and incentives needed to motivate the community), as discussed by other authors, are considered. In this paper, we discuss observations on our approach to improve the quality and sustain community ontology refinement though the use of social gaming and interaction. Current observations show that profile and knowledge of the concept in question, understanding and expressivity of the relationships play a key role in the quality of the result.

Keywords: Matcher selection, self-organization, community driven ontology engineering, social gaming, incentives.

1 Introduction

Given the disadvantages of expert driven ontology engineering [1], communities of stakeholders would have to be involved in the engineering process to allow the capture of emergent data and concepts and keep pace with ontology evolution. Social Networks are suitable for this process as the members all share common background knowledge, goals, and interests**.** Researches by [2][3][4][6] have explored community-driven approaches or games-with-a-purpose to develop the ontology. However, these works do not consider the familiarity and perception of the user and their ability to provide quality feedback (with regards to the concept in question). Also, the motivation or incentive for the user to continuously provide input to the system and its sustainability in terms of application propagation and social influence is not fully considered. Our methodology is to present the engineering task as a Facebook game to verify a lightweight ontology extracted from delicious and consider common background knowledge or familiarity to concept. Different aspects of social influence (both direct and indirect) [5] are used to allow for sustainability and scalability of the system. In terms of sustainability, we refer to direct influence such as friend requests to participate, while for indirect influence, we look at general awareness of peer activities via public postings. For scalability, the community is allowed to engineer the ontology and version the ontology on a community basis (small world graphs) and it uses direct and indirect peer influence to allow self monitoring and propagation of the application. For both cases, activities that promote social influence are tightly integrated to incentive schemes to motivate the community to perform those actions.

2 Results and Observations

During our two-month testing with 110 users subdivided into 4 community groupings, we encountered several issues and discovered some interesting results. It was seen that groups who are familiar with the area (topic) were able to participate more. These were also the users who, generally, return more than once; as opposed to the community that is from a different background. This result serves as a promising response to the assertion that selection of participants is important and knowledge of the tag in question by the community is necessary for them to provide feedback. Also, users are more capable of identifying erroneous relationships as opposed to validating them. The results show that the majority of the players “disagreed” with the initial discovered ontology. Perception and interpretation affect how concepts will be organized. Limiting our current discussion to simply identifying hierarchical relationships such as “kind-of”, it can be seen that connections validated and invalidated differ between groups. For the non-computer science community, some users perceived video as a “kind-of” learning mechanism (when they meant to express “videos can be used for learning”). This was due to the lack of expressivity of the relationships in the game. It should be noted that no “kind-of” relationship between “video” and “learning” was produced from inputs in the computer science user community. It was observed perspectives and interpretation play key roles and user profile and social affinity can serve as some of the bases to limit and select users for participation. Finally, the ontology is able to stabilize and converge; however, it is still not known whether it is the stability of the domain that allowed the ontology to converge or the process itself.

References

1. Dieter Fensel. Ontology Management Semantic Web, Semantic Web Services, and Business Applications. [ed.] Pieter De Leenheer, Aldo de Moor, York Sure Martin Hepp. New York : Springer Science+Business Media, LLC, 233 Spring Street, New York, NY, 2008.
2. Siorpaes, Katharina, Prantner, Kathrin and Bachlechner, Daniel. Class Hierarchy for the e-tourism Ontology Version 8. e-Tourism. [Online] November 2004. [Cited: January 6, 2009.] http://e-tourism.deri.at/ont/.
3. Cardoso, Jorge and Sheth, Amit P. Semantic Web Services, Processes and Applications. United States of America : Springer Science+Business Media, LLC, 2006.
4. Steve LEUNG, Fuhua LIN, Dunwei WEN. Towards a Data-driven Ontology Engineering. Taipei : The 16th International Conference on Computers in Education, 2008.
5. Eytan Bakshy, Brian Karrer,Lada A. Adamic. Social Influence and the Diffusion of User-Created Content. California : ACM, 2009. pp. 325-334. ISBN:978-1-60558-458-4.
6. Hepp, Katharina Siorpaes and Martin. OntoGame: Weaving the Semantic Web by Online Gaming. Tenerife : Springer, 2008.