Proceedings of the Symposium

Symposium on Matching and Meaning

A symposium at the AISB 2009 Convention (6-9 April 2009)
Heriot-Watt University, Edinburgh, Scotland

Symposium Chairs
Fiona McNeill

Published by SSAISB:
The Society for the Study of Artificial Intelligence
and the Simulation of Behaviour
http://www.aisb.org.uk/

ISBN - 1902956842

Workshop on Matching and Meaning

A one-day symposium at AISB 2009 (6-9 April 2009).
http://dream.inf.ed.ac.uk/events/wmm-2009/

PROGRAMME CHAIRS

Dr Fiona McNeill, University of Edinburgh, UK

INTRODUCTION

The problem of semantic misalignment - that of two systems failing to understand one another when their semantic representation is not identical - occurs in a huge variety of areas: the Semantic Web, databases, natural language processing; anywhere, in fact, where semantics are necessary but centralised control is undesirable or impractical.

The advantages of semantic fluidity clash with the communication difficulties this fluidity leads to. It is therefore essential to develop tools to facilitate automated development or evolution of ontologies as it becomes apparant that the existing representation is insufficient or inappropriate for the task at hand, and for interpreting the links between seemingly disparate ontologies.

These problems are often addressed offline, assuming that full information about all concerned ontologies is available. However, in highly dynamic domains, where interactions are between a large, diverse and evolving community, it is not practical to manually pre-align all concerned ontologies, nor possible, usually, to have complete access to all such ontologies. Such integration must be done dynamically and automatically.

This workshop brings together researchers interested in the problems of automated development, evolution and interpretation of ontologies in the many different domains in which it occurs. The workshop is focussed on the exchange of ideas and the stimulation of debate and is intended to be a forum for researchers to present ongoing work and ideas, and to engage in discussion with other researchers from the field.

TOPICS

Topics of interest include but are not limited to:

- Ontology evolution
- Ontology matching and alignment
- Ontology versioning
- Representational or structural change
- Formal aspects of ontology dynamics
- Foundational issues
- Social and collaborative matching
- Background knowledge in matching
- Extensions to ontology languages to better support change
- Belief revision for ontologies and the Semantic Web
- Inconsistency handling in evolving ontologies
- Uncertainty in matching
- Change propagation in ontologies and metadata
- Ontologies for dynamic environments
- Dynamic knowledge construction and exploitation
- Case studies, software tools, use cases, applications
- Open problems

PROGRAMME COMMITTEE

Manuel Atencia Arcas, IIIA-CSIC, Spain
Paolo Besana, University of Edinburgh, UK
Alan Bundy, University of Edinburgh, UK
Jerome Euzenat, INRIA Grenoble Rhone-Alpes, France
Fausto Giunchiglia, University of Trento, Italy
Adam Pease, Articulate Software, USA
Pavel Shvaiko, TasLab, Informatica Trentina, Italy

Table of Contents

Example-assignment to WordNet Thesaurus based on Distributional Similarity of Words

Fumiyo Fukumoto and **Nik Adilah Hanin Binti Zahri**†
Interdisciplinary Graduate School of Medicine and Engineering
Univ. of Yamanashi, 4-3-11, Takeda, Kofu, 400-8511, Japan
fukumoto@yamanashi.ac.jp **g07mk019@yamanashi.ac.jp**†

Abstract. In this paper, we present a method for assigning example sentences to each sense of words in WordNet. The key idea is that the method assigns each sense of a word w collected from not only the sentences containing w, but also sentences containing words with semantically related to w. Because a collection of a $context$ of the target word w is a similar syntactic behavior, even collected from a very very large corpora. The evaluation result showed that example-assignment based on groups of similar words significantly improved the retrieval of context for word sense, which helps to determine the exact definition of polysemous words.

1 Introduction

Word Sense Disambiguation (WSD) is one of the important problems in computational linguistics, as it is necessary at one level or another to accomplish most NLP and their applications [20]. One of the major approaches to disambiguate word senses is supervised learning [6], [18], [17]. A typical algorithm constructs a training set from all contexts of a polysemous word in the lexical resources such as machine-readable dictionaries or text corpora, and uses it to learn classifier that maps instances of the polysemous word into the senses [19]. A large number of papers published in this area involve comparisons of different learning approaches trained and tested with commonly used corpora or dictionaries. Unfortunately, not all the semantics are made explicit within lexical resources, even WordNet, the most widespread computational lexicon of English. Moreover, there are not a large amount of example sentences for each sense of words which are used to train classifiers. The production of semantically richer lexical resources can help alleviate the ontology acquisition bottleneck and potentially enable advanced NLP applications. However, in order to reduce the high cost of manual annotation, and to avoid the repetition of this effort for each lexical resource, this task must be supported by wide-coverage automated techniques which do not rely on the specific resource at hand.

In this paper, we present a method for assigning example sentences to each sense of words in WordNet. The key idea is that the method assigns each sense of a word w collected from not only the sentences containing w, but also sentences containing words with semantically related to w. Because a collection of a $context$ of the target word w is a similar syntactic behavior, even collected from a very very large corpora [2]. Here, a $context$ of the target word w is any sentence that contains w in the corpus. Consider the word "book" in WordNet. It has at least five senses including $account$ sense. We would use, in addition to the contexts of "book", all the contexts of "account" in the Reuters'96. The word "book" has a sense "account" when it co-occurred with the word "balance". However, co-occurrences between "book" and "balance" are not observed even in a very large corpus, while those between "account" and "balance" are high. For instance, in a corpus collected from one month Reuters, the number of sentences that contains "book" and "balance" were only 6, while those of "account" and "balance" were 65. Therefore, if we can find that "book" and "account" are semantically related, and collect contexts containing "account" and "balance", it would be possible to improve disambiguation accuracy using only seed annotated sentences *i.e.*, example sentences in WordNet together with a large unlabeled corpus, without requiring any additional hand labeling.

2 Overview of the System

The method consists of two steps: collecting semantically similar words and sentence retrieval. Figure 1 illustrates an overview of the system.

2.1 Collection of Semantically Similar Words

The first step to assign example sentences to WordNet thesaurus is to collect similar words from a corpus. A typical measure of similarity between words is based on their distributional similarity [9], [3]. Similarity measures based on distributional hypothesis compare a pair of weighted feature vectors that characterize two words. Features typically correspond to other words that co-occur with the characterized word in the same context. It is then assumed that different words that occur within similar contexts are semantically similar. Lin proposed a word similarity measure based on the distributional pattern of words which allows to construct a thesaurus using a parsed corpus [12]. We used it to calculate word similarities. More precisely, the similarity between two words is measured by the ratio between the amount of information needed to state the commonality of two words and the information needed to fully describe what the two words are [11]. It is based on their grammatical relationship with other words in the text corpus. We used Lin's syntactic parser to extract dependency triples [10]. The example of dependency triples is shown in Table 1.

The amount of information of a word *w* consists of all dependency triples that matches the pattern (*w*,*,*), where wild card (*) indicates frequencies including all the dependency triples that matches the particular pattern. Let the notation $\|w, r, w'\|$ represents the frequency count of dependency triples (*w*, *r*, *w'*). $\|cook, obj, *\|$, for example, defines the frequency counts of cook-object relationship , and

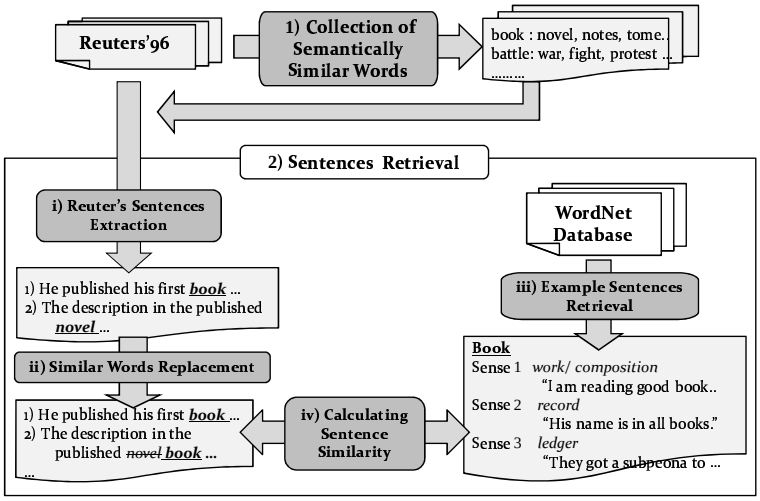


Figure 1. Overview of the System

Table 1. Example of dependency triples

Sentence:	He published his first book in 2006
Dependency triples:	(publish, subj, he), (publish, obj, book) (book, gen, his), (book, post, first) (book, mod, in), (in, pcomp-n, 2006)

$\|\|*,*,*\|\|$ defines the total frequency of dependency triples extracted from the parsed corpus.

The similarity of two words is measured based on the frequency of dependency triples. An occurrence of dependency triple (*w, r, w'*) is composed by the following three co-occurrence events:

A : randomly selected word, *w*

B: randomly selected dependency type, *r*

C: randomly selected word,*w'*

The probability of A,B and C co-occurring is estimated by

$$P_{MLE}(B)P_{MLE}(A|B)P_{MLE}(C|B),$$

where

$$\begin{aligned} P_{MLE}(B) &= \frac{\| *,r,* \|}{\| *,*,* \|}, \\ P_{MLE}(A|B) &= \frac{\| w,r,* \|}{\| *,r,* \|}, \\ P_{MLE}(C|B) &= \frac{\| *,r,w' \|}{\| *,r,* \|}. \end{aligned} \quad (1)$$

P_{MLE} is the maximum likelihood estimation of a probability distribution. When the value of $\|w,r,w'\|$ is known, we can obtain $P_{MLE}(A,B,C)$ directly:

$$P_{MLE}(A,B,C) = \frac{\| w,r,w' \|}{\| *,*,* \|} \quad (2)$$

Let *I*(*w,r,w'*) denotes the amount information contain in $\|w,r,w'\|$=c and can be computed as:

$$\begin{aligned} I(w,r,w') &= -logP_{MLE}(B)P_{MLE}(A|B)P_{MLE}(C|B) \\ &\quad -(-logP_{MLE}(A,B,C)), \\ &= log\frac{\| w,r,w' \| \times \| *,r,* \|}{\| w,r,* \| \times \| *,r,w' \|}. \end{aligned} \quad (3)$$

Let $T(w)$ be the set of pairs (r,w') such that $log\frac{\|w,r,w'\|\times\|*,r,*\|}{\|w,r,*\|\times\|*,r,w'\|}$ is positive. The similarity of two words, $SIM(w_1,w_2)$ is defined as:

$$\frac{\sum_{(r,w)\in T(w_1)\cap T(w_2)} (I(w_1,r,w)+I(w_2,r,w))}{\sum_{(r,w)\in T(w_1)} I(w_1,r,w)+\sum_{(r,w)\in T(w_2)} I(w_2,r,w)}. \quad (4)$$

A WordNet thesauri entry is created by using Eq. (4). For each noun word (we call it *seed* word), we extracted the top-5 words with the highest similarity value as a group of similar words.

2.2 Sentence Retrieval

The second step is to retrieve example sentences from Reuters, and assign each sentence to each sense of words in the WordNet. This step consists of two sub-steps: similar words replacement and calculating sentence similarity. In the similar words replacement, we replaced all the similar words in the extracted sentences from Reuters with seed word. The purpose of this sub-step is to increase the frequency of one-to-one correspondence words between WordNet and Reuters, which will extend the value of sentences similarity in the next sub-step. Next, we calculate two sentences from Reuters and WordNet examples by using formula (5).

$$Sent\_sim(W_i,R_i) = \frac{co(W_i \times R_i)+1}{\mid W_i \mid + \mid R_i \mid -2co(W_i \times R_i)+2},$$

where

$$\mid X \mid = \sum_{x\in X} f(x),$$

$$co(W_i \times R_j) = \sum_{(wn,reu)\in W_i \times R_j} min(f(wn), f(reu)),$$
$$W_i \times R_j = \{(wn, reu)|wn \in W_i, reu \in R_j\}. \quad (5)$$

$f(x)$ denotes the frequency of x in the sentence X. In Eq. (5), W_i and R_j refer to a set of words of the i-th Reuters sentence and j-th WordNet example sentence, respectively. (wn, reu) refers to one-to-one correspondence between the words wn and reu by looking up the same word in both sentences or words within the same group. If the similarity value exceeded a certain threshold value, R_i is regarded having the similar sense of the corresponding word with an example sentence of W_i.

3 Evaluation

3.1 Data

We used Minipar [10], a broad-coverage English parser, to parse 1 year of Reuters'96 data from August 20th, 1996 to August 19th, 1997. These corpus contained 806,791 articles consisting of 9,026,595 sentences. We collected the frequency of dependency triples by Minipar and used them to collect similar words. Here, we only performed clustering of similar noun words. From 806,791 Reuters articles, we extracted *object* and *subject* grammatical relationship of dependency triples. From 289,239 of *object* and *subject* related pairs, we obtained 30,953 pairs of triple dependency that occurred at least 100 times. Next, we retrieved 3,167 nouns with frequency 1,000 or higher, and then performed clustering of similar words against them. For each noun, we created a thesauri entry which contains the top-5 words that are most similar to the $seed$ word by using the similarity measure mentioned in Section 2.1.

In order to perform sentences similarity measure against Reuters sentences, we used example sentences in WordNet[2] [14]. There are 11,473 example sentences extracted from WordNet Database Version 3.0. However, in sentence retrieval procedure, we used only 608 WordNet and 180,394 of Reuters sentences.

3.2 Collection of Similar Words

From 3,167 group clustered, we randomly selected 25 percent of groups: 792 groups obtained from similar words clustering to be evaluated them manually. We checked if each word belongs to the corresponding groups. The sample of evaluation for the groups of similar words is shown in Table 2. The bold font word indicates that the word does not belong to its group.

We can see from Table 2 that some groups such as "willingness" and " obligation" were perfectly clustered, while "north" group consists of different sense of words. Table 3 shows the distributional of 792 groups. "# number of words correctly clustered" refers to the number of words that actually belong to the corresponding group determined by the system, and "# number of groups" denotes the amount of group evaluated with corresponding number of correctly determined similar words. Table 3 shows that the total number of similar words correctly identified by the system was 1,315, which resulted precision value, P= 0.332.

3.3 Sentence Retrieval

We evaluated 1,629 sentences with similarity that exceeded the threshold value, θ = 0.3. The list of words and the amount of sentences evaluated are listed in the Table 4. "baseline" in Table 4 shows

[2] available at http://wordnet.princeton.edu/obtain

Table 2. Samples of word clustering evaluation

Nouns	Similar words
access	use, control, link, **privatization**, **standard**
acceptance	recognition,**integration**, **creation**, **efficiency**, **participation**
accord	pact, treaty, agreement, **legislation**, **measure**
council	commission, committee, parliament, cabinet, agency
corruption	crime, abuse, violation, violence, disease
criticism	**threat**, **speculation**, complaint, **allegation**, comment
discussion	negotiable, debate, **privatization**, investigation, study
disorder	infection, **outbreak**, illness, **epidemic**, **cancer**
fluctuation	appreciation, downturn, swing, slump, instability
holiday	break, **start**, **auction**, **session**, **entry**
match	game, championship, round, race, **final**
north	**province, island, town, west, district**
regulation	rule, legislation, law, standard, pact
view	expectation, statement, term, report, idea
willingness	desire, determination, readiness, intention, commitment

Table 3. Distributional of evaluation results

# of Words Correctly Clustered in a Group	# of Groups	Total # of Correctly Clustered Words
0	210	0
1	192	192
2	182	364
3	106	318
4	69	276
5	33	165
Total	792	1,315

the results obtained by sentence similarity measure which does not involve in word replacement procedure.

As can be seen clearly from Table 4, the precision value for baseline was higher, which was P = 0.828 against our method, P = 0.770. However, the number of sentences measured by our method was 1,254 sentences, which were definitely 8 times higher than the number of sentences retrieved by the baseline, 159 sentences. Moreover, we found that our method retrieved sentences for the same sense of any top-5 similar words which did not exist in WordNet database. Table 5 shows an example taken from the sentences for the word "*accord*" with *settlement* and *agreement* sense. We can see from Table 5 that our method retrieves 61 sentences that contain "pact", 54 sentences containing "treaty", and 21 sentences that contain "legislation" with the same sense of the word "accord". Some of the example sentences are:

Reu960826-18900: *There was an iron **pact** at work here between certain political party.*
Reu961018-22517: *The **pact** must be qualitative, and not quantitative.*
Reu970507-18324: *Detail of the **pact** is not immediately available.*
Reu961007-2549: *The **treaty** was nonetheless being observed by all party.*
Reu970512-25674: *"the **treaty** is flexible enough," he said.*
...

We also found that some sentences are too general and did not show any specific feature to differentiate multiple sense of corresponding word. Consider the following sentences taken from the re-

Table 4. Evaluation result of sentence similarity

Word	Sense by WordNet	The number of sentences evaluated			
		Baseline		Proposed method	
		Correct	Total	Correct	Total
access	approach	0	0	2	2
	act of entering	0	0	3	4
accord	agreement, settlement	31	33	688	834
admission	admittance, entry	19	25	250	351
disaster	catastrophe, ruin, mess, misfortune	7	7	24	29
	tragedy, calamity (events cause by nature)	1	1	3	3
discussion	give-and-take word	10	10	18	21
	treatment, discourse	0	0	2	4
drill	exercise, practice	19	43	99	169
fluctuation	a wave motion	7	7	18	22
penalty	punishment, sentences	61	62	108	151
regulation	rule (principle or condition)	3	3	36	36
royalty	bonus, commission	1	1	3	3
Total		159	192	1,254	1,629
Precision		0.828		0.770	

Table 5. Example of sentences generation

	Word	Top-5 similar words					Total
	accord	pact	treaty	agreement	legislation	measure	
Example sentences (WordNet)	2	**0**	**0**	6	**0**	8	16
# of sentences	31	**61**	**54**	463	**21**	58	688

sults.

Reu970203-9425: *This would only create an enormous **fluctuation**.*
Reu970312-22347: *This be a short-term **fluctuation**.*
Reu970312-22347: *It was just a normal **fluctuation**.*

According to WordNet Thesaurus, the word "*fluctuation*" is either a physical wave motion or unsteady and changing condition. These three sentences have both senses, and consequently some of the sentences retrieved did not have the target sense in the sentences.

4 Previous Work

Bootstrapping methods for automatically sense-tag a training corpus has been an interest, as it helps knowledge acquisition bottleneck, i.e., manual sense-tagging of a corpus. The earliest work in this direction are those of Hearst [8], Schütze [16] and Yarowsky [18]. Yarowsky's method resolves the problem of knowledge acquisition limitation faced by word-specific sense discriminators disregard the polysemy issues. The identification of rarely occurred word sense in corpus also successfully performed by using statically word-specific models. Gale *et al*. proposed the use of bilingual corpora to avoid hand-tagging of training data. English-French parallel aligned corpus is used to automatically determine sense of each word in target language [6]. One problem with this approach is that the techniques heavily rely on availability of parallel corpora, while the sizes as well as the domain of existing bilingual corpora are limited. Dagan proposed a similar method, but instead of a parallel corpus use two monolingual corpora and a bilingual dictionary [5]. This solves the problems of availability of parallel corpora, since monolingual corpora are much easier to obtain than parallel corpora.

More recently, language scientists and technologists are increasingly turning to the Web as a source of language data, because it provides a great big body of linguistic data [18], [1], [13]. Agirre *et al*. proposed a method to acquire training examples by using two publicly available corpora including Semcor and an additional corpus automatically acquired from the Web [1]. They reported that the accuracy using the Web data was decrease, especially when Web examples whose word sense did not appear in publicly corpus. Moreover, the problem of data sparseness, which is especially severe for work in WSD occurred. There are at least three techniques to solve the problem of data sparseness: smoothing, class-based, and similarity-based methods. Smoothing method is used to get around the problem of infrequently occurring events, and in particular to ensure that non-observed events are not assumed to have a probability of zero. The best-known smoothing method is Turing-Good [7]. Class-based model is a method to obtain the best estimates by combining observations of classes of words considered to belong to a common category. Resnik used the taxonomy of WordNet and Yarowsky used the categories of $Roget's Thesaurus$ to define classes [15], [18]. Similarity-based method is to estimate similar words of the target word by using some similarity metric between patterns of co-occurrence [4]. Our methodology, especially word replacement procedure uses similarity-based method which makes it possible to assign sentences not containing the target word w to each sense of w.

5 Conclusion

Reliable retrieval of example sentences of word sense from text corpus opens up many approaches in the future especially for machine translation and information retrieval systems. This paper presented

the initial step to the resolution of lexical semantic ambiguity or known as WSD. In the context of WSD, our methods retrieved large number of example sentences for each sense. The experimental results showed that the number of sentences retrieval by group of similar words was 1,254 sentences, which was 8 times higher than baseline method, 159 sentences. The main contribution of this paper is a new method to retrieve sentences for word senses automatically with minimum test data or sentences used for comparison. Our method expands the use of automatic constructed thesauri and helps to develop sentences retrieval for WSD. Moreover, we found that our method retrieved sentences for the same sense of any top-5 similar words which did not exist in WordNet database. Future work will include (i) applying the method to other thesaurus such as Roget's thesaurus and LDOCE, and (ii) applying the method to WSD task.

REFERENCES

[1] E. Agirre and D. Martinez, 'Exploring automatic word sense disambiguation with decision lists and the web', in *Proceedings of the 18th International Conference on Computational Linguistics*, pp. 11–19, (2000).

[2] M. Banko and E. Brill, 'Scaling to very very large corpora for natural language disambiguation', in *Proc. of the 39th Annual Meeting and 10th conference of the European Chapter*, pp. 26–33, (2001).

[3] I. Dagan, L. Lee, and F. C. N. Pereira, 'Similarity-based models of word cooccurrence probabilities', *Machine Learning*, **34**(1-3), 43–69, (1999).

[4] I. Dagan, S. Marcus, and S. Markovitch, 'Contextual Word Similarity and Estimation from Sparse Data', in *Proc. of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 164–171, (1993).

[5] I. Dagan, F. Peireira, and L. Lee, 'Similarity-based Estimation of Word Cooccurrence Probabilities', in *Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 272–278, (1994).

[6] W. A. Gale, K. W. Church, and D. Yarowsky, 'Using bilingual materials to develop word sense disambiguation method', in *Proc. of the International Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 101–112, (1992).

[7] I. J. Good, 'The Population Frequencies of Species and the Distribution of Population Parameters', in *Biometrika*, pp. 237–264, (1953).

[8] M. A. Hearst, 'Noun homograph disambiguation using local context in large corpora', in *Proceedings of the 7th Annual Conference of the Centre for the New OED and Text Research: Using Corpora*, pp. 1–22, (1991).

[9] D. Hindle, 'Noun classifi cation from predicate-argument structures', in *Proc. of 28th Annual Meeting of the Association for Computational Linguistics*, pp. 268–275, (1990).

[10] D. Lin, 'Principar—an effi cient broad-coverage principle-based parser', in *Proceedings of the 15th International Conference on Computational Linguistics*, pp. 42–48, (1994).

[11] D. Lin, 'Using Syntactic Dependency as Local Context to Resolve Word Sense Ambiguity', in *Proc. of 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 64–71, (1997).

[12] D. Lin, 'Automatic Retrieval and Clustering of Similar Words', in *Proc. of 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pp. 768–773, (1998).

[13] R. Mihalcea, 'Using wikipedia for automatic word sense disambiguation', in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL2007)*, pp. 196–203, (2007).

[14] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, 'Introduction to wordnet: An online lexical database', in *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, pp. 112–119, (1990).

[15] P. Resnik, 'WordNet and Distributional Analysis: A Class-based Approach to Statistical Discovery', in *Proceedings of the AAAI Workshop on Statistically-based Natural Language PRocessing Techniques*, pp. 48–56, (1992).

[16] H. Schütze, 'Dimensions of Meaning', in *Proc. of Supercomputing'92*, pp. 787–796, (1992).

[17] M. Stevenson and Y. Wilks, 'The interaction of knowledge sources in word sense disambiguation', *Computational Linguistics*, **27**(3), 321–350, (2001).

[18] D. Yarowsky, 'Word sense disambiguation using statistical models of roget's categories trained on +arge corpora', in *Proc. of the 14th International Conference on Computational Linguistics*, pp. 454–460, (1992).

[19] W. Yorick, D. Fass, C. M. Gao, J. E. McDonald, T. Plate, and B.M. Slator, 'Providing machine tractable dictionary tools', *Machine Translation*, **5**(2), 99–154, (1990).

[20] W. Yorick and M. Stevenson, 'The grammar of sense: Is word sense tagging much more than part-of-speech tagging?', in *Technical Report CS-96-05(University of Sheffield)*, (1996).

Relating small ontologies

Alan Smaill and **Markus Guhe** and **Alison Pease**[1]

1 Introduction

We are interested in exploring representation change in mathematics; work by [5] suggests that the structure of metaphor plays an important cognitive role in the development of mathematical theories. In their work, metaphor is taken to involve "grounded, inference-preserving cross-domain mappings" [5, p 6].

One way to think about such mappings builds on work by Goguen [2, 1], who proposed related notions of (semiotic and frame) *morphism* to express relations that can hold between some statements or signs making a statement about some domain, and some other statements or signs related to a different domain.

One case is where the statements may be in a simple ontology language, but where there is a series of related ontologies, each describing different but also related domains. Goguen's *frame morphisms* are suggestive in giving a framework for describing the components of this situation.

2 A Case Study

To illustrate the approach, we look at the central example in Lakatos's famous reconstruction of the history surrounding Euler's formula [4]. It is of course interesting to try to model the evolving theories that crop up during the refinement of the ideas involved: Lakatos's suggestions allow certain basic operations of theories to be computationally realised, e.g. as described in [6]. Here, however, we are interested in analysing what goes on in the original, flawed, *procedural* proof, given in terms of a set of steps intended to preserve certain properties.

Steps in the argument involve carrying out operations on a system of connected faces, considered to lie on a flat plane. A sequence of miniature ontologies describes the state of this system at different stages.

The approach via frame morphisms suggests that we relate each ontology to a geometrical (or perhaps combinatorial) object. An operation on the ontology, e.g. consisting of *the removal of a point and two lines* can be defined over the syntax of the ontology. There should then be a related morphism, in the opposite direction, between the corresponding geometrical objects: in this case the simple embedding of one system of faces into another extended system works well.

The ontology statements can be taken to be statements in first-order logic. The semantics involved is not the standard Tarskian reading, however, since we are invoking the notion of a single canonical model. An advantage of the approach via frame morphisms is that we are not tied to a particular logic, and can thus happily make use in this way of aspects of closed world reasoning which are natural in this context.

[1] University of Edinburgh, email: {A.Smaill,M.Guhe,A.Pease}@ed.ac.uk

The subsequent history of the mathematics of this example involves the field of Algebraic Topology, which is full of "inference-preserving cross-domain mappings" that relate the algebraic and topological domains. The analysis above brings out aspects of the reasoning involved in going from a given 3-dimensional polygon to the more combinatoric graph-like reasoning involved elsewhere, and we can claim that the intuitions involved in this example are grounded in manipulations of packing cases, and so on. Related work in diagrammatic reasoning, such as [3], suggests that such forms of reasoning are found more intuitive than conventional syntactic proofs. An open question here is to what extent such grounding might be involved in the much more abstract developments of Algebraic Topology.

REFERENCES

[1] Joseph Goguen, 'An introduction to algebraic semiotics, with application to user interface design', in *Computation for Metaphors, Analogy, and Agents*, volume 1562 of *Lecture Notes in Computer Science*, pp. 242–291. Springer, (1999). doi:10.1007/3-540-48834-0_15.

[2] Joseph Goguen, 'What is a concept?', in *Conceptual Structures: Common Semantics for Sharing Knowledge*, eds., Frithjof Dau et al., volume 3596 of *Lecture Notes in Artificial Intelligence*. Springer, (2005).

[3] M. Jamnik, *Mathematical Reasoning with Diagrams: From Intuition to Automation*, CSLI Press, Stanford, CA, 2001.

[4] I. Lakatos, *Proofs and Refutations: The Logic of Mathematical Discovery*, Cambridge University Press, 1976.

[5] George Lakoff and Rafael Núñez, *Where Mathematics Comes From*, Basic Books, 2000.

[6] A. Pease, S. Colton, A. Smaill, and J. Lee, 'Modelling Lakatos's philosophy of mathematics', in *Proceedings of the Second European Computing and Philosophy Conference, E-CAP2004*, University of Pavia, (2004).

From "Baby Stuff" to the World of Adult Science: Developmental AI from a Kantian viewpoint.

Aaron Sloman [1]

Abstract. In contrast with ontology developers concerned with a symbolic or digital environment (e.g. the internet), I draw attention to some features of our 3-D spatio-temporal environment that challenge young humans and other intelligent animals and will also challenge future robots. Evolution provides most animals with an ontology that suffices for life, whereas some animals, including humans also have mechanisms for *substantive* ontology extension based on results of interacting with the environment. Future human-like robots will also need this. Since pre-verbal human children and many intelligent non-human animals, including hunting mammals, nest-building birds and primates can interact, often creatively, with complex structures and processes in a 3-D environment, that suggests (a) that they use ontologies that include kinds of material (stuff), kinds of structure, kinds of relationship, kinds of process and kinds of causal interaction and (b) since they don't use a human communicative language they must use information encoded in some form that existed prior to human communicative languages both in our evolutionary history and in individual development. Since evolution could not have anticipated the ontologies required for all human cultures, including advanced scientific cultures, individuals must have ways of achieving substantive ontology extension. The research reported here aims mainly to develop *requirements* for explanatory designs. Developing forms of representation, mechanisms and architectures that meet those requirements will have to come later.

1 INTRODUCTION: THE PROBLEM

This is an incomplete set of notes prepared for the AISB'2009 workshop on Matching and Meaning, on 9th April 2009 `http://dream.inf.ed.ac.uk/events/wmm-2009/`.

Current machine perceptual and manipulative abilities are extremely limited compared with what humans and many other animals can do, despite vast amounts of effort that have gone into work on vision, learning, planning, reasoning and robotics. I try to show that the reasons for the inadequacies are not at all obvious and to indicate some research directions that may be worth pursuing in order to bridge the gaps. A major part of the research task is identifying requirements to be met.

Any information processing system that interacts with some portion of reality by acquiring and making use of information about that reality, needs an ontology if it is to be able to acquire and use new information. It needs an *extendable* ontology if it is to be able to acquire and use new *kinds* of information. What this means is not easy to explain.

An ontology can be explicit or implicit. It is explicit if the contents are specified in some formalism that can be manipulated, stored, transmitted, used to make inferences, etc. (Intermediate cases would have only a subset of these capabilities.)

The ontology is implicit if there is no such formal specification, only a set of mechanisms that deal with instances of the ontology. For example, a common thermostat uses an implicit ontology in which there are temperatures that can vary continuously in one dimension and a control circuit that is either on or off. A washing machine controller has an implicit ontology which allows different washing programmes to be selected, a programme to be started, running or finished, and to go through a sequence of states while running. The designers and human users will make use of an explicit ontology, but the machine has no idea what users are doing or thinking, or what it has done or could do.

It is very likely that most of the ontologies used in most animal brains are implicit. Humans are an exception, at least for some parts of the ontology, and there may be other animals with explicit ontologies. Future human-like robots will probably need explicit ontologies for some of their activities. The designers of such robots will almost certainly need explicit ontologies (and meta-ontologies) even if the things they design use only implicit ontologies.

Biological evolution is a designer with only implicit ontologies and meta-ontologies – unless something is encoded in genomes that nobody has discovered.

An (explicit or implicit) ontology is an (explicit or implicit) specification of what sorts of things can be referred to or represented, what kinds of larger configurations they can be part of, what kinds of things can be part of them, what sorts of properties and relationships can categorise them, in what sorts of ways they can change (i.e. what processes can occur) what kinds of things can be known about them, what kinds of information can be missing about them, what sorts of inferences can be made, and how information about them can used in interacting with them.

An ontology need not be discrete, so it need not be representable with a tree-structured taxonomy: types of existent may be spread out in continuous spaces. Entities may be capable of existing in different sets of relationships, e.g. temporal, spatial, causal, functional, social, and economic, and therefore an ontology can be a tangled high-dimensional network – or more likely it will have a structure for which the notion of dimension is not relevant since its complexity instead of everywhere being factored into N independently variable components may be different in different parts of the network. (This is true of spaces defined by grammars, for example: the set of sentences in English does not have a dimension. Neither does the set of biological organisms.)

If the ontology is extendable, allowing for the discovery, or conjecture, of previously unknown types of reality, then it will not have a fixed structure. A system that can extend its ontology may have

[1] University of Birmingham, UK, http://www.cs.bham.ac.uk/~axs/

an explicit meta-ontology, specifying the ways in which the ontology can change or develop. Alternatively the meta-ontology may be implicit in mechanisms that perform those changes.

Investigating what sort of ontology, fixed or changeable, with or without an explicit meta-ontology, is required for an animal, human, or robot of a particular sort to function in our world is a hard research problem. At present I cannot specify the form of ontology in any detail, so this paper will be very vague in some respects. It is hoped that it will not be too vague to drive some directed research towards removing the vagueness.

2 OTHER KINDS OF ONTOLOGY RESEARCH

Many researchers, some of them labelled as "ontology engineers" are attempting to develop ontologies, or mechanisms for automatically generating ontologies, for use in connection with symbolic or digital environments, such as the internet or a set of corporate databases in a company or government organisation. (E.g. see[6] [8]) In contrast, this paper is not about the development, management, or use of artificial ontologies, but about ontologies used in certain animals. If we can understand the requirements to be met by such ontologies and can find good ways of creating and using them in artificial systems, then perhaps, at some future date, we shall be able to create robots that develop and use them, thereby overcoming one of the serious obstacles to producing robots with human-like intelligence. It could also inform educationalists, and reduce the obstacles schools and other institutions present to development of humans with human-like intelligence.

It may also turn out that biologically inspired ontologies and ontology-related mechanisms are also needed if machines are to understand many of the contents of the internet and other information stores in the same ways as humans do – since most of the contents of the internet are created by humans and are understood by humans, using their biological information-processing systems. So there is no apriori reason why any machine should be able to develop an appropriate understanding without using ontologies and mechanisms that are similar (at some level of abstraction) to those used by humans. That is left as an open question in this paper.

I should make it clear that I do not yet know how to design systems that meet the requirements discussed in this paper, and I don't think anyone else does either. However, in the long run, the common practice of starting from things we know how to do and then looking for minor variants may not, on its own, lead to significant progress, any more than looking for your lost keys where the lamplight is. However, combining that with research on more detailed requirements specifications may eventually reveal new ways to make progress.

3 WHAT ARE ONTOLOGIES FOR?

In many systems for which ontologies are being developed by software engineers, the purpose of the ontology is to facilitate a collection of symbolic operations, e.g. translating documents from one formalism or format to another, or to support data-mining operations in documents and databases from different sources. However, those applications are designed to interact only with symbolic structures, addressing problems that are very different from the problems that confront some organisms (or future robots) interacting with a 3-D environment. The software may produce graphical displays to help human users, e.g. showing a tree or graph on a screen, but typically the machine does not perceive the display, and works only on a symbolic description of the display. Even if it could see and reason about the visual display, that would not be the same as seeing, reasoning about and manipulating 3-D structures and processes in a physical environment.

There are some AI vision and robotic systems that perform impressively in very restricted 2-D or 3-D physical task domains, requiring little understanding of what they are doing or why it works. Examples include balancing a pole, repeatedly welding identical car bodies, or assembling components on a production line.

Some mobile robots are very impressive as hardware+software engineering products, e.g. BigDog – the Boston dynamics robot `http://www.bostondynamics.com/content/sec.php?section=BigDog` and some other mobile robots that are able to keep moving in fairly rough terrain, including, in some cases, moving up stairs or over very irregular obstacles, and, in the case of BigDog, recovering automatically from being pushed off balance, by sticking out a leg to prevent a fall.

However, like a river that very successfully gets water from mountains to the sea, they lack understanding of what they are doing, what they have done, what they could have done, what goals they could achieve in different circumstances, why some goals should be abandoned, etc. though they can sometimes react *as if they* understood, either because a programmer designed the control software to act appropriately, or because a training regime caused a learning mechanism to adjust the control parameters to perform as needed. It is possible to combine a variety of programmed or trained condition-action rules that together give the appearance of understanding, but hide underlying rigidity – like a paint spraying robot that uses only previously learnt movements to spray a piece of furniture even if confronted with an item whose shape is different and needs different spray movements.

Such machines cannot, and have no need to, explain or think about what they are doing, why they are doing it, whether there is any other way of doing it, why they are not doing that, etc. They cannot watch someone else doing similar things and make suggestions for avoiding errors or improving performance. They cannot hypothesise that something is happening that they have never previously encountered that may require their ontology to be extended.

Existing robots that manipulate objects can be triggered to perform an action, but cannot perceive processes, notice new possibilities, or reason about what the result would be if something were to happen, except in very simple cases.

Neither can they reason about why something is not possible. I.e. they lack abilities to perceive and reason about positive and negative affordances.

They cannot wonder why an action failed; wonder what would have happened if they had done something differently; notice that their action might have failed if so and so had occurred part way through, etc.; or realise after the event that some information was available that they did not notice at the time.

Those robots have implicit ontologies. An agent with an implicit ontology has mechanisms that are driven by what happens and in what circumstances, which causes chains of reactions that lead to behaviours responding to those happenings in those situations. In the case of many organisms, e.g. microbes, insects and many others, this type of restricted competence based on an implicit ontology is adequate for the species to continue existing – even if many individuals die "prematurely" because of the rigidity of their responses.

But not all organisms are like that. Some of them (certainly humans, and arguably some others) have the ability to generate information structures representing things that do not exist, including processes that could occur, situations that could arise, things that might

have happened in the past, and things that the agent hypothesises might exist, but cannot be perceived either because they are too far away, or obscured by intervening matter, or because they cannot be detected by the agent's sensors, even though when present they can, under some conditions, have effects that are perceived – like the invisible molecular structure that causes sugar, but not sand, to dissolve in water when stirred.

The ability to generate such hypothetical information structures involves having a store of meanings (concepts) that can be combined in various ways to represent possible objects or object parts, states of affairs, events or processes – without having to be driven by sensory inputs. This requires having some *form of representation*, namely something like a *medium* that can be interpreted as expressing information, and which allows meaningful structures to be constructed, manipulated, stored, combined, disassembled, and used for various purposes.

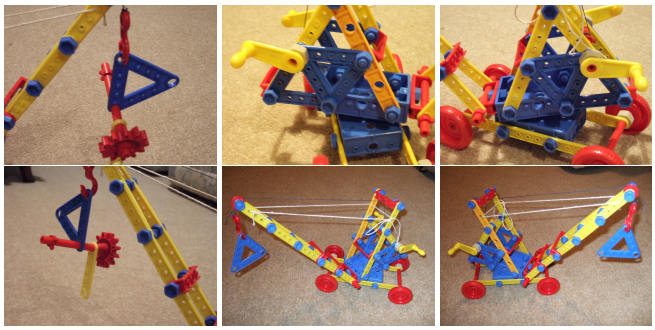
Some of the flexibility of explicit ontologies comes from the fact that the expressive medium need not be physical: in many cases it is more convenient to use symbols or representations composed of entities in virtual machines [17] [14] [12] [9]. Putting such pieces of meaning together is a kind of creativity, which provides the ability to deal with novel situations as well as the ability to represent non-existent situations. (Compare Boden [1] and [11, Chap 2].)

4 ONTOLOGIES FOR SEEING

It is not always noticed that perception can involve similar creativity, in unfamiliar situations. The possession of a suitable ontology, with appropriate generative forms of representation able to express possibilities within the ontology, is required for seeing a novel situation, insofar as such perception involves creating a new usable information structure, either transiently or, if the situation is remembered, in a medium or long term information store. (Compare understanding a sentence you have never heard or read previously, like some of the sentences here.)

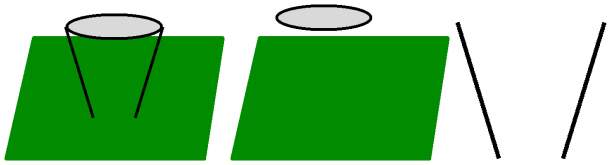
Familiarity with roles of low level pictorial cues in representing 3-D edges, orientation, curvature of surfaces, joins between two objects or surfaces, etc., allows you to use compositional capabilities to see 3-D structure, and some causal and functional relationships, in pictures (even static, monocular pictures) never previously seen.

How many features, relationships (topological, semi-metrical, metrical, causal) can you see in these pictures, taken from `http://www.cs.bham.ac.uk/research/projects/cosy/photos/crane`?

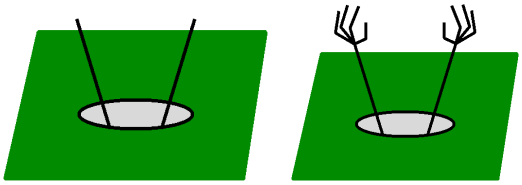


No AI vision system comes close to being able to see these.

The combinatoric creativity involved in perceiving and understanding spatial structures is very different from that involved in doing algebraic or logical operations, because whereas individual components of an algebraic or logical formula retain their syntactic and semantic functions when the elements are rearranged, or when the formula is embedded in a larger formula, that is not true of "analogical" representations of spatial configurations, as pointed out in [10] and [11, Chapter 7].



For example what you see in the above pictures? Only 2-D configurations? Or do you see them as involving 3-D structures and relationships. Do you interpret the same things always in the same way, or are the 3-D relations you see between two parts of the scene dependent on what else is in the image? Notice how context can influence interpretation of parts. Perceptual compositional semantics is highly context-sensitive, as shown by comparing the interpretation of the image components in the previous pictures with the interpretations below. What does the picture on the right suggest to you? Obviously that will depend on how the two multi-pronged structures are interpreted.



Words can add more context that influences how the image is interpreted, by activating different parts of your ontology of 3-D structures, processes, relationships. The figure on the right was based on a "droodle" found on the internet with the caption "Strong worm catches early bird". Another possible caption is "Shark-infested sewer". Different people asked to invent captions will produce different phrases, depending on which ontologies they have and also how fragments of their ontologies are activated by the fragments and relationships of fragments in the picture.

5 SEEING POSSIBILITIES

Some of the interpretations of the droodle above depend on hypothesising physical structures that are not visible but are connected to the structures that are visible. Interpreting droodles, however, would not normally be regarded as a typical illustration of biological functions of vision. In contrast, the images below, which are poor quality, low resolution photographs of 3-D configurations, can be interpreted as scenes in which various kinds of actions are possible, where the possibilities are constrained by the perceived structures – taken from this presentation: `http://www.cs.bham.ac.uk/research/projects/cogaff/challenge.pdf`

If you attend to various locations on the surfaces of these objects and if you see their 3-D shapes, even with low precision and much noise,

you will be able to work out roughly how two fingers need to be oriented to grasp at those locations. For instance, try attending to, or getting someone to point at various parts of the rim of the cup, or of the edge of the saucer, or of the spoon or cup handle. In each case you should be able to work out roughly how your finger and thumb will need to be oriented if you grasp at that point. In doing that you are making use of an ontology of spatial positions, orientations and relationships that can hold between surfaces or between objects with surfaces. This shows that perception of action affordances makes uses of an ontology allowing a whole range of spatial arrangements between surfaces of fingers and other objects, among other things. The word "roughly", earlier, indicates that the locations and orientations are not represented with great precision – an important feature of the ontology.

You can probably go beyond thinking about static grasping, to construct a representation of a *process* in which you transform a configuration of one sort into the other sort – which could be done in various different ways, depending where things are held, where they are placed in intermediate configurations, etc. That is, you can make a plan in your head for transforming the configuration seen in one picture of the cup saucer and spoon the configuration in the other picture.

In doing that visualisation, you use an ontology of process fragments that can be "attached" to the various portions of the visible surfaces of the objects involved. Doing that is working out a plan: this need not be a very metrically precise plan in order to be the basis of intelligent action, or intelligent advice to someone else about how to rearrange the objects. An important question that I shall not try to answer in any detail is how you manage to steer through the explosive (infinite) space of continuously varying possible movements at various levels of abstraction so as to find a particular set of movements (a plan) that achieves the desire result. It is often assumed that this requires a search through a discrete space of possible combinations of actions, whose discreteness results from the ability to chunk continuous spaces into sub-spaces (often with slightly fuzzy boundaries). Exactly how that space is related to the ability to see is a topic for another time.

It is not necessary to recognize any of the object categories for the purpose of constructing such a plan, as long as you can see their shapes, namely how the various parts of their surfaces are arranged in space. I could have used pictures of objects that you did not recognise at all. You can do many things with something you see but do not recognize, including planning things to do to it.

6 EXTENDABLE ONTOLOGIES

Extendable, and at least partly explicit, ontologies are needed by animals that have to acquire and use information about, reason about, and interact with rich and complex 3-D structures and processes in the physical environment. Even perception of 2-D images of representing abstract structures, such as written words, requires use of ontologies with multiple layers using different sub-ontologies, as illustrated in [11, Chap. 9]. Impressive recent work in machine vision has shown how computers can use statistical methods to automatically *induce* ontologies as a result of being exposed to many pictures, e.g. the work by Fidler and colleagues in [4, 3]. However at present such ontologies are concerned (a) only with static structures and (b) levels of 2-D organisation in images, as opposed to being able to cope with processes in 3-D space with changing 3-D structures and relationships. Perhaps the methods used will generalise to those cases (using considerably more computing power).

However, not only do the ontologies used in visual systems need to be extendable to cope with new types of entity, they also need to be usable for more tasks than recognition or description of externally presented configurations. As illustrated earlier, the ontology needs also to be usable for representing and reasoning about non-existent but *possible* entities and processes, including possible future sequences of events, but also for representing things in the past, or out of sight, or invisible but capable of having visible effects.

Similar requirements are relevant to future machines doing automated design, inspection and repair of complex machinery; automated rescue systems; domestic aids for disabled people; and robots performing tasks in remote and humanly uninhabitable environments, e.g. on space platforms and other planets. For such systems, required ontologies will not refer only to abstract structures (e.g. web pages and their contents, collections of scientific data, or business information systems concerned with financial transactions) but also to some of the sorts of things many animals can deal with, including spatial structures and processes, causal interactions, assembly or disassembly of objects of varying degrees and kinds of complexity, including changes of

- material properties (e.g. becoming brittle),
- spatial relations (including shape changes),
- causal relations (e.g. producing obstructions, or loosening a grip)
- functional relations (e.g. modifying a structure to serve a new purpose)

7 WITH WHAT STARTING POINT?

A newborn human infant cannot see or do all those things. Why not? – And what has to change to produce those competences? It seems that newborn humans start off with a limited ontology provided by evolution [7] along with the ability to extend the ontology by interacting with the environment.

Some newborn animals can do very sophisticated things very soon after birth (e.g. deer, chicks) so evolution **can** produce innate sophisticated competences, with whatever ontology is required. However in many cases an implicit ontology will suffice.

Infant humans, orangutans, corvids, ... lack behavioural competences some other species have at birth or hatching, even though the other species do not develop so far in their lifetime. Perhaps that is because humans are born with something more powerful than the competences picked up by other animals. This is the hypothesis under development in the collaboration reported in [13, 2].

Many researchers assume *learning* is that more powerful something: but what sort of learning? And from what starting point?

A common assumption is that the initial learning is of a general kind, that can learn anything, provided that enough training data can be provided.

The designers of such systems don't bother to study the environment: they expect to leave that to their future learning systems – but that may not work, for the reason given by McCarthy in [7]:

> "Evolution solved a different problem than that of starting a baby with no a priori assumptions.
> .....
> Instead of building babies as Cartesian philosophers taking nothing but their sensations for granted, evolution produced babies with innate prejudices that correspond to facts about the world and babies' positions in it. Learning starts from these prejudices. What is the world like, and what are these instinctive prejudices?"

A logicist roboticist might think all the required innate prejudices can be expressed as axioms and deployed through a logic engine. However, studying the environment animals interact with, and learn in, suggests that we need a much richer theory, involving what McCarthy describes, and also

- An initial architecture, that can extend itself in certain ways, including ontology extension.
- Initial (still unknown) forms or representation adequate for encoding specific sorts of information (including information about processes in which 3-D surfaces change their shapes and spatial relations), and which support specific forms of information manipulation.
- Initial sensory, motor, and internal processing mechanisms, including mechanisms for constructing new goals, for goal conflict resolution, and for detecting opportunities to learn.
- Initial behavioural dispositions that drive learning tailored to perceiving and producing 3-D structures and processes.
- An initial, mostly implicit, "framework theory" determining the type of ontology that is assumed and ways in which it can be used and extended. Compare Kant's [5].

  E.g. implicit assumptions about the topology of space/time, kinds of stuff able to occupy and move around in space, modes of composition of structures and processes, kinds of process that can occur involving the stuff, kinds of causation, the differences between doing and passive sensing, ...
- Delayed activation of an architectural layer that uses the combination of the environment and the early architecture as a new developmental "playground" in order to drive ever more sophisticated testing, debugging, and extensions as conjectured in [2].

8 WHAT SORT OF INITIAL ONTOLOGY?

Many theorists assume that the initial ontology includes only sensory and motor contents and patterns relating them, a somatic, multimodal, ontology) – I claim that will not suffice for children, chimps, or crows. Instead, I conjecture that from the start many learners will use, and attempt to extend, an *exosomatic*, amodal ontology (about what's going on outside – not just the shadows on Plato's cave wall), including:

- bits of stuff (of various kinds) that can occur in the environment
- bits of surface of bits of stuff, in various shapes, locations, orientations
- bits of process (of various kinds) that can occur in the environment
- ways of combining them to construct larger structures and processes in the environment (not necessarily with global consistency)
- at various levels of abstraction: metrical, semi-metrical, topological, causal, functional....

Semi-metrical representations include things like: "W is further from X than Y is from Z", orderings with gap descriptions, symmetries and partial symmetries. (And other things, still to be determined.)

Semi-metrical distance and angle measures could include comparisons between distances and angles instead of use of global units, like 'cm' or 'degrees'.

Instead of items in the environment being located relative to a single global coordinate frame, they could be embedded in (changing) networks of more or less local relations of the above types.

9 HOW CAN ALL THIS WORK ?

Powerful multi-layer, extendable constraint-propagation mechanisms will need to be available for vision, haptic perception, reasoning, planning, predicting, etc. to work. For more on this see, for example, [18]. The main unsolved problem seems to be: what forms of representation are required to support these processes?

It is argued in [16] that in our pre-linguistic evolutionary history, our pre-verbal individual development and in some other non-verbal animals, there are "languages" that are not used for communication, but are used **internally** for perception, reasoning, goal formation, planning, plan execution, question formation, prediction, explanation, causal understanding, as described above, and those languages include (a) structural variability (for dealing with novelty), (b) compositional semantics (modified by context sensitivity) (c) manipulability (for reasoning, planning, hypothesising, etc.).

Suggestions for making progress

Instead of the normal AI strategy of thinking about how to extend our existing mechanisms, or how to deploy them in new ways, perhaps we should spend more time engaged in a deep study of features of the environment and ways of interacting with it, looking at examples of children and other animals doing that, and altering their competences as a result. On that basis we can try to derive constraints on the forms of representation and ontologies that can explain the detailed phenomena observed at different stages of development (which in children are *partially*, not *totally* ordered).

In the light of all that, we should try to design and test mechanisms, architectures, robots that illustrate the theories.

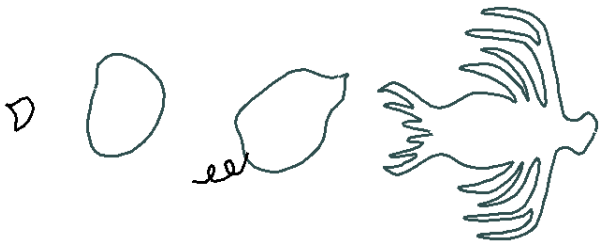
The problems will be different for different sorts of organisms and robots, e.g. depending on the complexity of their sensors and manipulators, the kinds of terrain they inhabit and the kinds of things they need to acquire and avoid. See:[15]

Composition/binding

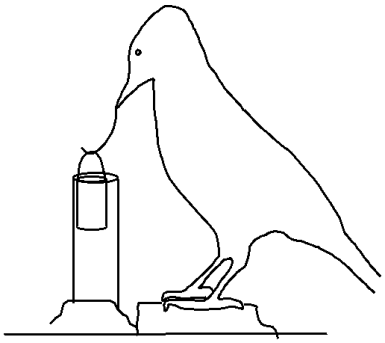
These different aspects of reality can be composed/combined in many different ways. Long before there was algebraic/functional/logical composition there was spatio-temporal composition. Also auditory/temporal composition – in music and many natural sounds. We need to distinguish composition in the spatio-temporal environment (e.g. combining actions and things acted on, or sounds) from composition in internal representations of things that can be spatio-temporally combined (e.g. composition in representations in virtual machines).

At present we have only a relatively small number of forms of information-composition that we can implement and use in computers. Perhaps by studying the environments of various sorts of intelligent systems very carefully we can derive new requirements for forms of representation and forms of composition and manipulation. This may lead to the creation of new kinds of artificial information-processing systems.

10 LIFE IS INFORMATION PROCESSING



The world contains matter, energy, and information. Organisms acquire and use information, in order to control how they use matter and energy – in order to acquire more matter, energy and information, and also reproduce, repair, defend against intruders, dispose of waste products... Somehow evolution produced more and more sophisticated information processors, driven in part by changes in the environment, which led to changes in morphology which provided more opportunities in the environment requiring more sophisticated information processing (for example, when organisms acquired manipulators that could move independently of eyes), as conjectured in [15]. It seems that evolutionary advances driven/selected by particular challenges often produced opportunities for new more complex advances.



Betty, the New Caledonian Crow made hooks from straight pieces of wire, in several different ways, in order to get a bucket of food out of a tall transparent tube.

All this poses great challenges for science and engineering, namely, to understand that process, to understand the products, and to design working systems that replicate various aspects of the products. In order to do this we need a better understanding of

- the structure of design space
- the structure of niche space
- the many design tradeoffs linking them
- the possible trajectories in design space,
- the possible trajectories in niche space,
- the many complex feedback loops linking both.

11 DEVELOPMENT OF ENVIRONMENT AND COGNITION

The cognitive system, including sensory mechanisms, motor control systems, learning systems, motivational mechanisms, memory, forms of representation, forms of reasoning, etc. that an organism (or robot) needs will depend both on what is in the environment and also what the physical structure and capabilities of the organism are. The current fashion for emphasising the role of embodiment in cognition mostly leads to claims that a particular form of embodiment *solves* or *eliminates* cognitive problems. My claim, on the contrary is that added complexity of animal bodies provides new more complex problems of cognition and control, as explained in [18].

For a micro-organism swimming in an ever changing chemical soup it may suffice to have hill-climbing mechanisms that sense and follow chemical gradients, perhaps choosing different chemical gradients according to the current needs of the organism.

As the environment becomes more structured, more differentiated with more enduring objects and features (e.g. obstacles, food sources, dangers, shelters, manipulable entities) and the organisms become more articulated, with more complex changing needs, the information-processing requirements become increasingly more demanding.

As more complex information processing capabilities develop, the opportunities to observe, modify and combine them in new ways also develop.

The cognitive system, including sensory mechanisms, motor control systems, learning systems, motivational mechanisms, memory, forms of representation, forms of reasoning, etc. that an organism (or robot) needs will depend both on

- what is in the environment
  and
- what the physical structure and capabilities of the organism are.

Many researchers who emphasise the importance of embodiment of animals and robots make a mistaken assumption:

> they claim that embodiment and physical morphology solve the problems and reduce the burdens on cognition, by producing required results "for free" when movements occur.

However, the point I am making is that as bodies become more complex, with more parts that can be moved independently to cooperate with one another in performing complex actions on complex, changeable structures in the environment, the cognitive demands (for perception, learning, planning, reasoning, and motor control, and the ontologies involved) increase substantially, requiring more powerful forms of representation and more complex information-processing architectures.

12 TURING'S MISTAKE?

A major challenge for such an investigation is to understand the variety of possible starting points for an individual born or hatched in a particular sort of environment, after millions of years of evolution of the species. In [19] Turing wrote:

> "Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child brain is something like a notebook as one buys it from the stationer's. Rather little mechanism, and lots of blank sheets. (Mechanism and writing are from our point of view almost synonymous.) Our hope is that there is so little mechanism in the child brain that something like it can be easily programmed."

On this point (little mechanism and much space), Turing was uncharacteristically badly wrong, like all the AI researchers who try to find a small number (some hope **one** will suffice) of powerful, general, learning mechanisms that can learn from arbitrary data: *Evolution did not produce general-purpose data-miners.*

Most species produced by evolution start off with almost all the information they will ever need, leaving only scope for minor adjustments of parameters, e.g. for calibration and minor adaptations. A few species learn a lot using mechanisms that evolved to learn in a 3-D world of static and changing configurations of objects, including other intelligent agents: they start with powerful special-purpose mechanisms. In short: *Evolution itself is a general-purpose data-miner, changing what it mines.* But it needs something like a planet-sized laboratory, and millions of years, to produce things like humans

13 MCCARTHY DISAGREES WITH TURING

As indicated earlier, John McCarthy, in [7], emphasises an important point missed by Turing (and by many AI researchers). In the same article he wrote:

> "Animal behavior, including human intelligence, evolved to survive and succeed in this complex, partially observable and very slightly controllable world. The main features of this world have existed for several billion years and should not have to be learned anew by each person or animal."

McCarthy's own theories about requirements for a neonate are tempered by his goal of attempting to see how much could be achieved using logic. We need to keep an open mind as to which forms of representation and modes of syntactic composition and transformation may be required, or may be useful at times. (As argued in 1971 in [10], and Chapter 7 of [11].)

I am not arguing **against** the use of logic, but **for** a search for additional (new) forms of representation.

14 DEVELOPMENTAL PSYCHOLOGISTS vs DESIGNERS

Many developmental psychologists investigate what is and is not innate in newborn humans, and other animals. Examples studying humans include (among many more): E. Spelke, P. Rochat, E. Gibson & D. Pick, A. Karmiloff-Smith, and much earlier J. Piaget,
and studying animals: N. Tinbergen, K. Lorenz, J. Goodall, W. Köhler, E.C. Tolman, I. Pepperberg, M. Hauser, A. Kacelnik (and colleagues), N. Clayton, S. Healey, F. Warneken, M. Tomasello. Unfortunately not enough of these researchers have learnt to look at something done by a child, chimp, or chick and ask

- How could **that** work? What else can the mechanisms do?
- **How** do they do it?

Instead many researchers ask questions like:

- Under what conditions does this happen?
- How can the task be made easier or more difficult for species X?
- Is this innate or learnt?
- If it is learnt what triggers the learning?
- Which other animals can and cannot do it?
- How early does it happen?
- Which additional tests can I perform to detect these and similar competences?

They don't adopt what McCarthy calls "the designer stance". That is a very difficult thing to do.

ACKNOWLEDGEMENTS

Several of these ideas resulted from interactions with members of the EU-funded CoSy robotic project `http://www.cognitivesystems.org` and its sequel the CogX project `http://cogx.eu` at the University of Birmingham and in partner organisations. Some of the points here referring to non-human animals were influenced by colleagues in the School of Biosciences working on animal cognition, Jackie Chappell and Susannah Thorpe. My thinking on these topics was inspired in part by reading Kant and Piaget many years ago.

Apologies for this unfinished/unpolished paper. Later versions will be available here:
`http://www.cs.bham.ac.uk/research/projects/cogaff/sloman-mm09.pdf`

REFERENCES

[1] M. A. Boden, *The Creative Mind: Myths and Mechanisms*, Weidenfeld & Nicolson, London, 1990.
[2] Jackie Chappell and Aaron Sloman, 'Natural and artificial meta-configured altricial information-processing systems', *International Journal of Unconventional Computing*, **3**(3), 211–239, (2007). http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0609.
[3] S. Fidler, M. Boben, and A. Leonardis, 'Similarity-based cross-layered hierarchical representation for object categorization', in *Proceedings Conference on Computer Vision and Pattern Recognition,*, pp. 1–8. IEEE Press, (2008). http://sanja.fri.uni-lj.si/wp-content/uploads/fidler08cvpr.pdf.
[4] S. Fidler and A. Leonardis, 'Towards Scalable Representations of Object Categories: Learning a Hierarchy of Parts', in *Proceedings Conference on Computer Vision and Pattern Recognition,*, pp. 1–8. IEEE Press, (2007). http://vicos.fri.uni-lj.si/data/alesl/cvpr07fidler.pdf.
[5] I. Kant, *Critique of Pure Reason*, Macmillan, London, 1781. Translated (1929) by Norman Kemp Smith.
[6] Catherine Legg, 'Ontologies on the Semantic Web', **41**, 407–452, (2007,).
[7] J. McCarthy, 'The well-designed child', *Artificial Intelligence*, **172**(18), 2003–2014, (2008). http://www-formal.stanford.edu/jmc/child.html.
[8] Olena Medelyan and Catherine Legg, 'Integrating Cyc and Wikipedia: Folksonomy meets rigorously defined common-sense', in *Proceedings, Workshop on Wikipedia and AI AAAI'08*, eds., R. Bunescu, E. Gabrilovich, and R. Mihalcea, pp. 13–18, Menlo Park, CA, (2008). AAAI.
[9] J. L. Pollock, 'What Am I? Virtual machines and the mind/body problem', *Philosophy and Phenomenological Research.*, **76**(2), 237–309, (2008). http://philsci-archive.pitt.edu/archive/00003341.
[10] A. Sloman, 'Interactions between philosophy and AI: The role of intuition and non-logical reasoning in intelligence', in *Proc 2nd IJCAI*, pp. 209–226, London, (1971). William Kaufmann. http://www.cs.bham.ac.uk/research/cogaff/04.html#200407.
[11] A. Sloman, *The Computer Revolution in Philosophy*, Harvester Press (and Humanities Press), Hassocks, Sussex, 1978. http://www.cs.bham.ac.uk/research/cogaff/crp.
[12] A. Sloman, 'The Well-Designed Young Mathematician', *Artificial Intelligence*, **172**(18), 2015–2034, (2008). http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0807.
[13] A. Sloman and J. Chappell, 'The Altricial-Precocial Spectrum for Robots', in *Proceedings IJCAI'05*, pp. 1187–1192, Edinburgh, (2005). IJCAI. http://www.cs.bham.ac.uk/research/cogaff/05.html#200502.
[14] A. Sloman and R.L. Chrisley, 'Virtual machines and consciousness', *Journal of Consciousness Studies*, **10**(4-5), 113–172, (2003).
[15] Aaron Sloman, 'Diversity of Developmental Trajectories in Natural and Artificial Intelligence', in *Computational Approaches to Representation Change during Learning and Development. AAAI Fall Symposium 2007, Technical Report FS-07-03*, eds., C. T. Morrison and T. Tim Oates, pp. 70–79, Menlo Park, CA, (2007). AAAI Press. http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0704.
[16] Aaron Sloman. Evolution of minds and languages. What evolved first and develops first in children: Languages for communicating, or languages for thinking (Generalised Languages: GLs)?, 2008. http://www.cs.bham.ac.uk/research/projects/cosy/papers/#pr0702.
[17] Aaron Sloman, 'Virtual Machines in Philosophy, Engineering & Biology', in *Proceedings Workshop on Philosophy & Engineering WPE-2008*, eds., Natasha McCarthy and David Goldberg, Royal Academy of Engineering, London, (2008). http://www.cs.bham.ac.uk/research/projects/cogaff/08.html#803.
[18] Aaron Sloman, 'Some Requirements for Human-like Robots: Why the recent over-emphasis on embodiment has held up progress', in *Creating Brain-like Intelligence*, eds., B. Sendhoff, E. Koerner, O. Sporns, H. Ritter, and K. Doya, 248–277, Springer-Verlag, Berlin, (2009). http://www.cs.bham.ac.uk/research/projects/cosy/papers/#tr0804.
[19] A.M. Turing, 'Computing machinery and intelligence', *Mind*, **59**, 433–460, (1950). (reprinted in E.A. Feigenbaum and J. Feldman (eds) *Computers and Thought* McGraw-Hill, New York, 1963, 11–35).

Converting Classifications into OWL Ontologies

Fausto Giunchiglia[1], Ilya Zaihrayeu[1], and Feroz Farazi[1,2]

[1]Department of Information Engineering and Computer Science
University of Trento, Italy
[2]Department of Computer Science and Engineering
University of Chittagong, Bangladesh
{fausto, ilya, farazi}@disi.unitn.it

Abstract. Classification schemes, such as the DMoZ web directory, provide a convenient and intuitive way for humans to access classified contents. While being easy to be dealt with for humans, classification schemes remain hard to be reasoned about by automated software agents. Among other things, this hardness is conditioned by the ambiguous nature of the natural language used to describe classification categories. In this paper we describe how classification schemes can be converted into OWL ontologies, thus enabling reasoning on them by Semantic Web applications. The proposed solution is based on a two phase approach in which category names are first encoded in a concept language and then, together with the structure of the classification scheme, are converted into an OWL ontology. We demonstrate the practical applicability of our approach by showing how the results of reasoning on these OWL ontologies can help improve the organization and use of web directories.

1 Introduction

A *classification scheme*, or a *classification* for short, is a rooted tree whose nodes are assigned natural language labels and are populated with a (possibly empty) set of documents. Since the invention of classification by Aristotle in the 4th century BC, classifications have been used (and are still used) pervasively to represent various kinds of human knowledge. For example, classifications have been used in libraries (DDC[1], LCC[2] and Colon classification[3]); in Personal Knowledge Management (favorites, personal e-mails and folder hierarchies); and, lately, on the Web (Amazon[4], Google[5], Yahoo[6]).

While classifications are heavily used to categorize web contents, the evolution of the web foresees a more formal structure which can serve this purpose

[1] See http://www.tnrdlib.bc.ca/dewey.html.
[2] See http://www.loc.gov/catdir/cpso/lcc.html.
[3] See http://www.iskoi.org/doc/colon.htm.
[4] See http://www.amazon.com.
[5] See http://www.google.com.
[6] See http://www.yahoo.com.

– *ontology*, defined in Computer Science as *a specification of a conceptualization* [10]. Ontologies are core artifacts of Semantic Web, an extension of the current Web, in which information is given formal semantics such that computers can use inference rules to conduct automated reasoning on pieces of this information [1]. The key factor which makes this possible is the fact that ontologies are expressed in a formal language, suitable for automated reasoning.

In this paper we bridge the gap between informal classifications and formal ontologies by describing an approach to encoding classification labels in a formal language such that, together with the structure of the classification scheme, they can be then converted into OWL [2] ontologies (more precisely, into lightweight ontologies, as described in [9]). In principle, the proposed approach allows for automated reasoning on classifications through reasoning on corresponding OWL ontologies. Moreover, the conversion is fully automated. Web directories can be encoded into OWL ontologies without user intervention. We demonstrate the practical applicability of our approach by showing how the results of reasoning on these OWL ontologies can help improve the organization and use of classification schemes. While encoding classifications into a formal language is not new, the main novelty of this paper consists of converting classifications into OWL ontologies, which demonstrates a proof of concept that classifications can be seamlessly integrated in the Semantic Web infrastructure. The fully automated algorithm described in this paper is also novel, as well the characterization of the expressivity of the formal language (i.e. OWL Lite, OWL DL, OWL Full) needed to encode classifications.

The rest of the paper is structured as follows. In Section 2 we describe a comparison between classification schemes and ontologies. In Section 3 we describe how to convert classification schemes into OWL ontologies and how the generated OWL ontolgies can be enriched with additional axioms. In Section 4 we report the experimental results. Section 5 presents how this work helps in optimizing classifications. In Section 6 we discuss the related work and we conclude the paper in Section 7.

2 Classification Schemes vs Ontologies

In this section we discuss commonalities and differences between classifications and ontologies. In order to ground our discussion on well defined terms, below we give the definitions of these two kinds of artifacts.

A *classification* is a 5-tuple $C = \langle N, E, L, D, cl \rangle$ where N is a finite set of nodes, E is a set of edges on N, such that $\langle N, E \rangle$ is a rooted tree; L is a finite set of labels expressed in natural language, such that for any node $n_i \in N$, there is one and only one label $l_i \in L$; D is a set of documents and cl is a function which maps every $d_i \in D$ to a non-empty set of nodes $\{n_i\} \subseteq N$. In Figure 1 we show an example of a classification. Although classifications have no explicit formal semantics for edges, in this example we labeled each edge with the name of a hypothetical relation that may hold between the linked nodes.

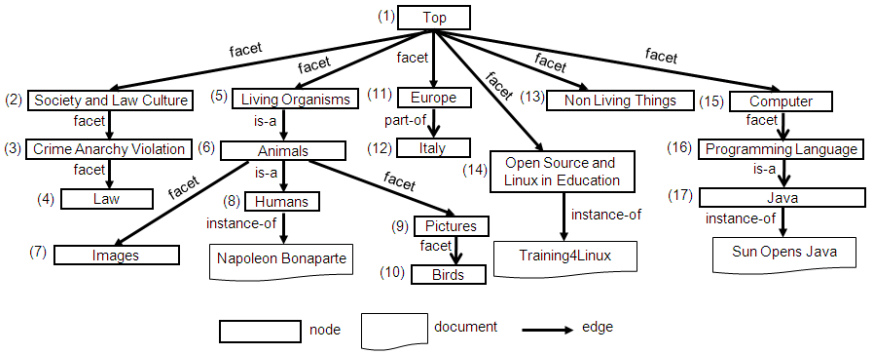


Fig. 1. An example of a classification with link semantics made explicit.

An *ontology* is an *explicit specification of a conceptualization* [10]. They are often thought of as directed graphs whose nodes represent *concepts* and whose edges represent formal *relations* between concepts. The backbone structure of the ontology graph is a taxonomy in which all the relations are `sub-class-of`, whereas the remaining structure of the graph supplies auxiliary information about the modeled domain and may include relations like `part-of`, `located-in`, `is-parent-of`, and others [11]. Classes can be associated with instances through the `instance-of` relation. In Figure 2 we show an example of a small ontology.

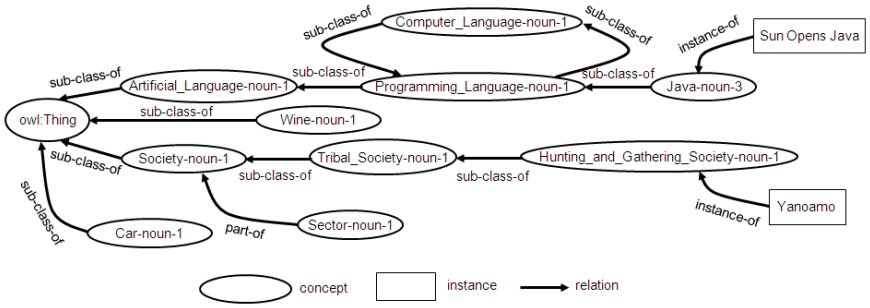


Fig. 2. An example of an OWL ontology.

Even if both ontologies and classifications can often be represented in the form of a graph, ontologies and classifications are quite different in their uses, purpose, language, applications, and in other aspects which we summarize as follows:

– **Users:** a typical user of classifications is a human (e.g., a classifier in a library classification), whereas ontologies are primarily used by machines and, as such, they are the key enablers of the Semantic Web. Moreover,

designing a classification is part of everyday practice of many computer users, whereas designing a full-fledged ontology (expressed, for example, in OWL-DL) is a difficult and error-prone task even for ontology experts [17];
- **Purpose:** classifications are primarily used for the organization of (large) document collections into categories and subcategories such that these documents can be easily accessed by a human through browsing the classification tree in a top-down fashion. Ontologies are primarily used for modeling a particular domain such that the resulting model represents a shared view of a group of individuals [16];
- **Language:** as from the definition, classifications use natural language to describe nodes' categories. Natural language is well understood by humans but, due to its ambiguous nature, it is hard to be "understood" and reasoned about by machines. In contrast, ontologies are codified in a formal language which is unambiguously interpreted by machines. In fact, because ontologies are expressed in a formal language, they are often used for automated *reasoning* about the domain they model. Natural language is used in ontologies in a limited extent (e.g., to describe concept names) and, in general, has no functional value in reasoning operations on ontologies;
- **Nodes:** in an ontology, nodes normally represent atomic concepts (e.g., `car`, `wine`), whose names are shown next to the corresponding nodes when ontologies are visualized. In a classification, a label can represent a rather complex concept (e.g., "Open Source and Linux in Education") or an individual (e.g., "Napoleon Bonaparte");
- **Edges:** in an ontology graph, edges have a well defined semantics and they usually encode `sub-class-of`, `part-of` and other relations that hold between the two concepts connected with by an edge. In a classification, an edge implicitly represents either: (i) a *specification* relation which can be thought of as an `is-a` relation (e.g., an edge from "Animals" to "Humans") or as a `part-of` relation (e.g., an edge from "Europe" to "Italy"); or, (ii) a *facet* relation which encodes the fact that the label of the child node represents an aspect of meaning of the parent node (e.g., an edge from "Animals" to "Images") [3]. It is a bad practice to connect two nodes whose labels denote disjoint concepts (e.g., "non-living things" and "living organisms") as in this case the child node and all its descendants cannot be populated with any document in a meaningful way;
- **Instances:** in an ontology, node instances are representatives of the node class and of all its ancestor classes in the `sub-class-of` hierarchy. They are in the `instance-of` relation with the class(es) they belong to. In a classification, node instances are not necessarily representatives of the class denoted by the node label, and can be documents which are about objects described by the set of labels of the nodes on the path from the given node to the root. For example, a node labeled "birds" may be populated with pictures of birds if the label of the parent node is "pictures".

As shown above, classifications and ontologies are quite different and they have their cons and pros with respect to each other. We summarize their dis-

tinguishable features in Table 1 and, in the next section, we show how we can bridge the gap between them thus combining their pros within a single knowledge representation structure.

Table 1. Comparison between classification schemes and ontologies

Category	**Classification Schemes**	**Ontologies**
Users	Humans	Machines
Purpose	Organization of (large) document collections	Modeling of a domain
Language	Natural language, e.g. English	Formal language, e.g. OWL
Nodes	Usually represent complex concepts or individuals	Usually represent atomic concepts
Edges	Do not have well defined semantics	Have well defined semantics
Instances	Are not necessarily instances of the class in which they are populated	Are instances of the class in which they are populated
Examples	DDC, LCC, Colon classification	Gene ontology[a], OpenCyc ontology[b], MeSH ontology

[a] http://www.geneontology.org/
[b] http://www.opencyc.org/

3 From Classifications to OWL Ontologies

In this section we show how a classification, as defined in Section 2, can be converted into an OWL ontology. Particularly, we show how classification elements, namely: labels, nodes, edges, documents, and document-node links are encoded into OWL structures. Note that encoding classification labels requires converting from a natural language to a formal language, whereas encoding classification nodes and edges requires only structural manipulation. In Section 3.1 we discuss how we solve the former problem and in Section 3.2 we show how we solve the latter one. In Section 3.3 we show how we encode classification documents and document-node links as class instances. In Section 3.4 we show how the resulting OWL ontology can be enriched with a set of axioms such that it can be better suited for automated reasoning. Finally, in Section 3.5 we discuss which subset of the OWL language is required in order to encode classifications into ontologies.

3.1 From Labels to Concepts of Labels

In the conversion of natural language labels into a formal language we follow the approach presented in [4], which describes how these labels can be converted into a propositional concept language. The underlying idea of this approach is that senses of words, appearing in a label, are converted into atomic concepts, whereas punctuation and syntactic relations between words in the label are converted into

logical connectives (such as conjunction ⊓ and disjunction ⊔) and parenthesis. As discussed in [9], the extension of these concepts is the set of documents about the objects or individuals referred to by the (lexically defined) concepts. As shown in the same article, this interpretation has some nice properties such as it provides the possibility to represent individuals as concepts, and not as instances (e.g., the extension of concept `George_Bush` is the set of documents about the president George Bush), and to treat classification edges as the intersection of concepts. In the analysis of natural language labels we exploit the natural language processing (NLP) pipeline presented in [21]. Differently from the standard approaches to NLP, this pipeline is adapted to be applied on web directory labels. In the following we present the main steps of the pipeline and we show how we complete some of them with the conversion to OWL.

1. Sense retrieval. At this step, we retrieve the senses of each word in the label from the WordNet lexical database [15]. Apart from this, we identify words which are not found in WordNet.

2. Sense disambiguation. At this step, we leave only one sense per ambiguous word following the word sense disambiguation algorithm presented in [21]. The algorithm exploits the structure of the classification, WordNet relations such as hypernymy, and the most frequent sense heuristic to disambiguate the meaning.

3. Building atomic concepts. At this step we convert the disambiguated senses as well as the words which are not found in WordNet into atomic concepts and encode them as OWL classes. Following the approach described in [19], we define the URI scheme to uniquely identify OWL classes generated from WordNet senses as follows:

`Synset- + lexical_form_of_the_word- + POS- + synset_number`

where `synset_number` is the number of the synset[7] to which the sense belongs in WordNet, and `lexical_form_of_the_word` is the lemma of the *first* word in the given synset. This allows us to represent synonymous words as one OWL class and not as multiple classes with equivalence relations defined between them.

For example, the URI for the atomic concept `java` which is generated from the sense *coffee* of the noun java is: `Synset-Coffee-Noun-41492`. An OWL class for this atomic concept is defined as follows:

<owl:Class rdf:ID="Synset-Coffee-Noun-41492" />

We form URIs for the words which are not found in WordNet as their literal representation in the label. For example, word *xyz* is encoded as an OWL class with URI "xyz". This allows us to encode unknown words with the same spelling as one OWL class. Note that the encoding of words into concepts is done in a way to build a minimal set of concepts. The main reason for this choice is efficiency.

4. Building complex concepts. At this step we build complex concepts from atomic concepts following the approach discussed in [4]. For instance, a label composed of a sequence of adjectives followed by a noun group is converted into the logical conjunction (⊓) of the concepts corresponding to the adjectives

[7] In WordNet, a *synset* is a set of one or more synonymous words which is assigned a unique numeric identifier, a gloss, and other metadata [15].

and to the nouns; prepositions like "of" and "in" are converted into the logical conjunction; coordinating conjunctions "and" and "or" are converted into the logical disjunction ($\sqcup$), and so on.

We convert complex concepts generated from the labels into classes in OWL. We define the following URI schema to uniquely identify these OWL classes:

`Label- + node_label- + node_number`

where `node_label` is the label of the node without spaces, and where each word starts with a capitalized letter. For example, the URI for the label "Society and Law Culture" of node 2 of the classification given in Figure 1 is: `Label-SocietyAndLawCulture-2`. The OWL class for this label is as follows:

```
<owl:Class rdf:ID="Label-SocietyAndLawCulture-2">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:ID="Synset-Society-Noun-318"/>
    <owl:intersectionOf rdf:parseType="Collection">
      <owl:Class rdf:ID="Synset-Law-Noun-51793"/>
      <owl:Class rdf:ID="Synset-Culture-Noun-38542"/>
    </owl:intersectionOf>
  </owl:unionOf>
</owl:Class>
```

3.2 From Concepts at Labels to Concepts at Nodes

As discussed in Section 2, edges in a classification represent either a specification or a facet relation, which can be generalized to the following observation: the meaning of a child node consists of what the meaning of its label and the meaning of the parent node have in common. We formalize this observation in the notion of *concept of node* [5, 8, 6, 7], which is defined as follows:

$$C_i = \begin{cases} l_i^F & \text{if } n_i \text{ is the root of } C \\ l_i^F \sqcap C_j & \text{if } n_i \text{ is not the root of } C, \text{ where } n_j \text{ is the parent of } n_i \end{cases} \quad (1)$$

where C_i is the concept of node n_i and l_i^F is the concept of label of node n_i. Concepts at nodes are converted into classes in OWL. The URI schema used to uniquely identify OWL classes corresponding to nodes is defined as follows:

`Node- + node_label- + node_number`

For example, in Figure 1 the URI for the root node labeled "Top" with id 1 is: `Node-Top-1`. An OWL class for this root node is built as follows:

```
<owl:Class rdf:ID="Node-Top-1">
  <equivalentClass rdf:resource="#Label-Top-1"/>
</owl:Class>
```

The URI for node 16 labeled "Programming Language" is: `Node-ProgrammingLanguage-16` and its corresponding OWL class is built as follows:

```
<owl:Class rdf:ID="Node-ProgrammingLanguage-16">
```

```
<owl:intersectionOf rdf:parseType="Collection">
  <owl:Class rdf:ID="Label-ProgrammingLanguage-16"/>
  <owl:Class rdf:ID="Node-Computer-15"/>
</owl:intersectionOf>
</owl:Class>
```

Note that classification edges are implicitly encoded in the definitions of OWL classes representing concepts at nodes. Namely, since these classes are defined as the intersection of the concept at node of the parent and the concept at label of the child node, then the structure of the classification can be reconstructed by analyzing node class definitions.

3.3 From Documents to Class Instances

We convert a document into an instance of the OWL Thing class. We assume that each document has a URL and we use it to uniquely identify the corresponding instance in OWL. Moreover, if a document has a title and a description (as web directory documents normally have), then we encode them in rdfs:label and rdfs:comment properties accordingly. For example, a document with URL `http://java-source.net/`, with title "Java Open Source Software", and with description "A directory of open source software focused on Java" is encoded in OWL as follows:

```
<owl:Thing rdf:about="#http://java-source.net/">
  <rdfs:label>Java Open Source Software</rdfs:label>
  <rdfs:comment>A directory of open source software focused on Java
  </rdfs:comment>
</owl:Thing>
```

We convert document-node links of a document by defining the rdf:type relation from the instance, representing the document, to the class(es) representing the node(s) in which the document is classified. For instance, if the above mentioned document is classified in nodes 2 and 4 of the classification shown in Figure 1, then these document-node links are encoded as follows:

```
<owl:Thing rdf:about="#http://www.laweasy.com">
  <rdf:type rdf:resource="#Node-SocietyAndLawCulture-2"/>
  <rdf:type rdf:resource="#Node-Law-4"/>
</owl:Thing>
```

3.4 Semantic Enrichment

Since OWL classes, which correspond to word senses, are mapped to synsets in WordNet, we can exploit the relations between synsets and relations between words within synsets in order to enrich the resulting OWL ontologies with additional relations between classes. The enrichment is based on these two rules:

– **Rule 1:** In WordNet, synsets are organized into hierarchies based, for example, on the hypernym (i.e., *is-a* or *is-kind-of*) relation [15]. For instance, the

synset denoting "Java" (as "a simple platform-independent object-oriented programming language") has a hypernym synset denoting "object-oriented programming language" (as "a programming language that enables the programmer to associate a set of procedures with each type of data structure"). If two OWL classes (`cl-1` and `cl-2`) correspond to two senses (`sen-1` and `sen-2`) belonging to two synsets (`syn-1` and `syn-2`) among which there is a hypernym relation defined in WordNet (e.g., `syn-2` is a hypernym for `syn-1`), then we define an `rdfs:subClassOf` relation between these two classes (i.e., `cl-1 rdfs:subClassOf cl-2`) as follows:

<owl:Class rdf:about="#Synset-Java-Noun-41493">
<rdfs:subClassOf rdf:resource="#Synset-ProgrammingLanguage-Noun-45-219"/>
</owl:Class>

– **Rule 2:** Antonym relations in WordNet are defined among *words* within synsets (and not among synsets). We translate these relations into `owl:disjointWith` relations among classes corresponding to senses of the two antonym words. For instance, the antonym of the word "day" in the synset {day, daytime, daylight} is the word "night" in the synset {night, nighttime, dark}. The former synset is the third sense of the noun "day" and the latter synset is the first sense of the noun "night". Classes, associated with these two senses, are declared to be disjoint as follows:

<owl:Class rdf:about="#Synset-Day-Noun-12826">
<owl:disjointWith rdf:resource="#Synset-Night-Noun-38819"/>
</owl:Class>

The enrichment of classification OWL ontologies according to the two rules described above allows us to make these ontologies more suitable for reasoning as the underline axiom base grows.

3.5 OWL Sublanguage

OWL ontologies, generated from classifications, fall into the OWL Lite or OWL DL subset of OWL. There are two factors which require OWL DL:

– the logical disjunction that may appear after the conversion of natural language labels and which is converted into the `owl:unionOf` construct;
– disjoint axioms that may appear at the semantic enrichment step and which are converted into the `owl:disjointWith` construct.

Both above mentioned constructs are forbidden in OWL Lite. Note that the conversion to OWL does not require the use of constructs of OWL Full which leaves us within a decidable subset of OWL.

4 Evaluation

To evaluate our approach, we selected four subtrees with the maximum depth of 3 from the DMoz web directory. In Table 2 we report statistical data of the datasets. There are 476 nodes in the selected subtrees, which have 548 tokens in total, out of which, 527 tokens are found in WordNet (i.e., WordNet coverage is 96.17%). Out of the set of words found in WordNet, 223 (i.e., 42.31%) are ambiguous with the average polysemy of 3.36. In our experiments we used WordNet version 2.0.

Table 2. Statistics of the dataset

Dataset	Nodes	Average Branching Factor	Average Subtree Depth	Tokens Per Label	Words with Senses in WordNet	Noun Senses	Adjective Senses
Countries[a]	245	6.26	3	1.07	261	256	5
Europe[b]	75	4.22	3	1.12	86	86	0
Asia[c]	76	4.24	3	1.18	89	88	1
Africa[d]	80	4.31	3	1.15	94	93	1

[a] http://dmoz.org/Regional/Countries/.
[b] http://dmoz.org/Regional/Europe/.
[c] http://dmoz.org/Regional/Asia/.
[d] http://dmoz.org/Regional/Africa/.

4.1 Correctness

We evaluated the most critical step of the NLP pipeline, i.e., the word sense disambiguation (see Section 3.1) algorithm, whose performance results are reported in Table 3. The accuracy of this step largely affects the correctness of the results of reasoning on these OWL ontologies, as we show in Section 5.5.

Table 3. Accuracy of the word sense disambiguation algorithm

Dataset	Ambiguous Tokens	Disambiguation Accuracy(%)
Countries	92	76.54
Europe	38	77.01
Asia	47	80.89
Africa	46	79.13

4.2 OWL Sublanguage

In Table 4 we report statistical data for the generated OWL ontologies and in Table 5 we provide details on the kind and number of axioms before and after semantic enrichment.

Table 4. Statistics of the generated OWL ontologies

Ontology	Nodes	Sense Classes	Label Classes	Node Classes	Class Axioms	Individual Axioms	intersectionOf Constructs	unionOf Constructs
Countries	245	261	245	245	873	0	265	4
Europe	75	86	75	75	155	183	76	10
Asia	76	89	76	76	203	125	80	9
Africa	80	94	80	80	212	253	84	9

Table 5. Axioms before and after semantic enrichment

Ontology	Equivalent Class Axioms		SubClass Axioms		Disjoint Class Axioms		Individual Axioms	
	Before	After	Before	After	Before	After	Before	After
Countries	490	490	0	383	0	0	0	0
Europe	152	152	0	3	0	0	183	183
Asia	152	152	0	51	0	0	125	125
Africa	160	160	0	52	0	0	253	253

Noteworthy, most of the constructs in the generated ontologies are valid in OWL Lite. There are only few `owl:unionOf` constructs, which require the use of OWL DL for the representation of these ontologies.

5 Optimizing Classifications

In this section we show some practical examples of reasoning on classification OWL ontologies. For instance, we show how they can be checked for consistency, how their structure can be rationalized, and how nodes with similar contents to a given node can be found.

5.1 Consistency

We used Protégé OWL Plugin [14] and its reasoning capabilities to detect logical inconsistencies within the classification OWL ontologies. We used reasoning capabilities of both Pellet 1.5 and Fact++ OWL reasoners launched with Protégé.

None of the reasoners reported that the classification OWL ontologies were inconsistent.

5.2 Rational Forms

Classifications may not be perfect. For this reason we may need to reconstruct a classification based on the "most specific subsumer" relation. Nodes get parents which most specifically describe them, still being more general. The new structure is called, a *rational form* of a classification. The idea behind the rationalization of classifications is to build a classification which better corresponds to a taxonomic structure. At the semantic enrichment step rational form is built. The classification given in Figure 3(b) is a rational form of the classification given in Figure 3(a). This rational form is obtained as "programming language" is a direct hypernym of "object-oriented programming language" in WordNet and that is computed and then converted to `owl:subClassOf` relation between them at semantic enrichment step. Note that classification semantics does not change when going from classification to rational form of classification as the set of concepts at nodes remains the same.

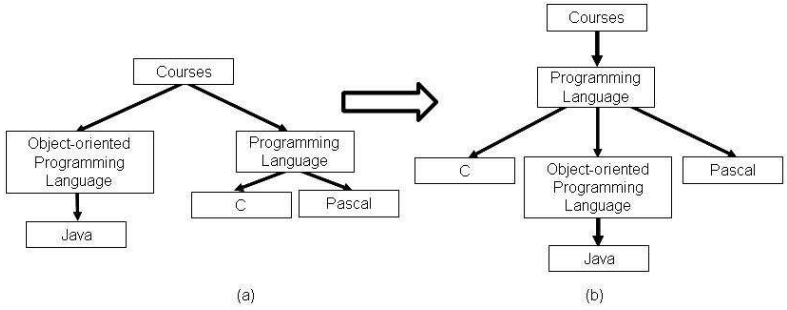


Fig. 3. (a) Classification; (b) Rational form of the classification given in (a)

5.3 Minimizing Effort

In Table 6 we report the kind and number of found relations within and across the four ontologies. For example, the reasoner found an equivalent relation between node class */Regional/Countries/**Italy*** and node class */Regional/Europe/**Italy***. This is an example of how reasoning on classification OWL ontologies can help web directory editors find interrelated parts of the web directory and, thus, improve its organizational structure without manual inspection. Note that no disjointness relations were found because we did not have disjoint axioms in the produced OWL ontologies.

Table 6. Found relations within and across the four ontologies

Ontology	Countries		Europe		Asia		Africa	
	$\sqsubseteq$	$\equiv$	$\sqsubseteq$	$\equiv$	$\sqsubseteq$	$\equiv$	$\sqsubseteq$	$\equiv$
Countries	383	490	386	642	392	642	387	650
Europe	386	642	3	152	51	304	52	312
Asia	392	642	51	304	51	152	100	312
Africa	387	650	52	312	100	312	52	160

5.4 Computing `See-Also` Links

Apart from the four ontologies, we experimented with another classification OWL ontology and we observed that the individuals asserted to the OWL class which corresponds to the classification node */Games and Activities/Kids and Teens/**Football*** are inferred as the individuals of the OWL class which corresponds to the classification node */Sports Athletics Funs/Youth and High School/**Soccer***, and vice versa. This kind of reasoning can be used for finding similar documents populated in different nodes, which will help in building `see-also` links.

5.5 Errors

Apart from correct relations, we found also some incorrect ones. For example, the reasoner found an erroneous more specific relation between node class */Regional/Europe/**Georgia*** and node class */Regional/Countries/**United States***. As discussed earlier, this problem is caused by the lack of accuracy of the word sense disambiguation algorithm. Evaluating the correctness and completeness characteristics of the computed set of relations between ontology classes is outside the scope of the current paper. Interested readers are referred to [5] for a complete account.

6 Related Work

The current work is a representative of a recent trend in the Semantic Web community towards the use of *lightweight semantics* (as opposed to expressive logic languages) and *lightweight ontologies* [9] (as opposed to full-fledged ontologies), the generation of which can be potentially supported by ordinary users which constitute the long tail of the Semantic Web. The trend has been formed through a number of scientific publications (e.g., see [20, 4, 18, 13]) and is currently supported by a number of R&D projects (e.g., MATURE[8], OpenKnowledge[9]) and systems (e.g., OntoWiki[10]). The current work contributes to this trend by proposing an approach in which classifications, which are often called

[8] MATURE, Integrated Project (IP), FP7-216356, see http://mature-ip.eu.
[9] OpenKnowledge, STREP, FP6-27253, see http://www.openk.org/
[10] OntoWiki, see http://ontowiki.net/Projects/OntoWiki.

(informal) lightweight ontologies [9] and whose most representative instantiations on the web are web directories, can be automatically converted into formal OWL ontologies, ready to be embedded in Semantic Web applications.

There are few lines of work which are close in spirit to our approach. For instance, in [20], the authors propose a method to converting thesauri to OWL ontologies in which they provide a detailed account of how elements of a thesaurus are converted into OWL structures. This approach is based on a manual analysis of thesauri, whereas our approach allows for a fully automatic conversion. Another approach, discussed in [18], comes from the Digital Library community and presents a conceptual structure and transition procedure to support the shift from a traditional knowledge organization system (KOS) and, particularly, a thesaurus, towards a full-fledged and semantically rich KOS. While providing an in-depth analysis of the shortcomings of the traditional KOSs and of the benefits of semantic KOSs as well as providing a set of rules for converting thesaurus elements into ontology constructs, the approach lacks a specification of how a KOS can be converted into an ontology language, such as OWL – the ultimate conversion step discussed in detail in the current paper.

The approach described in [12] allows us to convert a hierarchical classification into an OWL ontology by deriving OWL classes from classification labels and by arranging these classes into a hierarchy (based on the rdfs:subClassOf relation) following the classification structure. The approach is based on some application-dependent assumptions such as that one label represents one atomic concept, and that relations between labels can be defined as `sub-class-of` relations in some particular context (e.g., concept "ice" is more specific than concept "non-alcoholic beverages" when considered in the context of procurement). These assumptions do not hold in a general case and are not made in our approach. Apart from this, our approach differs from [20, 18, 12] in that it is generic and, therefore, suitable for automatic conversion in OWL of any knowledge representation structure whose core can be represented in the form of a classification as defined in this paper.

7 Conclusions

In this paper we have presented a fully automated approach to converting generic classification schemes into OWL ontologies. The proposed approach allows us to leverage on top of classifications, being the interfaces to knowledge for humans, and ontologies, being the interfaces to knowledge for machines on the Semantic Web. Furthermore, as shown above, our approach provides immediate advantage and it allows to help the user in building better classifications more suited for reasoning. Potentially, the approach allows for a cost-free seamless integration of a vast amount of classification structures on the web and in personal repositories into the Semantic Web infrastructure, thus reducing the problem of the lack of semantically rich data. The first experimental results, reported in this paper, show that reasoning on classification OWL ontologies can be used for building practical Semantic Web applications.

References

1. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, (284(5)):34–43, May 2001.
2. S. Bechhofer et al. OWL Web ontology language reference, W3C recommendation, February 2004.
3. F. Giunchiglia, M. Marchese, and I. Zaihrayeu. Towards a theory of formal classification. In *Proceedings of the AAAI-05 Workshop on Contexts and Ontologies: Theory, Practice and Applications (C&O-2005)*, Pittsburgh, Pennsylvania, USA, 2005.
4. F. Giunchiglia, M. Marchese, and I. Zaihrayeu. Encoding classifications into lightweight ontologies. In *Journal on Data Semantics (JoDS) VIII*, Winter 2006.
5. F. Giunchiglia and P. Shvaiko. Semantic matching. *Knowledge Engineering Review*, 18(3):265–280, 2003.
6. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Semantic schema matching. In *OTM Conferences (1)*, pages 347–365, 2005.
7. F. Giunchiglia and M. Yatskevich. Element level semantic matching. In *Meaning Coordination and Negotiation workshop, ISWC*, 2004.
8. F. Giunchiglia, M. Yatskevich, and E. Giunchiglia. Efficient semantic matching. In *ESWC*, pages 272–289, 2005.
9. F. Giunchiglia and I. Zaihrayeu. Lightweight ontologies. In *The Encyclopedia of Database Systems, to appear*. Springer, 2008.
10. T. R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, 1993.
11. N. Guarino. Some ontological principles for designing upper level lexical resources. In *First International Conference on Lexical Resources and Evaluation*, volume 2830, Granada, Spain, May 1998.
12. M. Hepp. Representing the hierarchy of industrial taxonomies in OWL: The gen/tax approach. In *Proceedings ISWC Workshop on Semantic Web Case Studies and Best Practices for eBusiness (SWCASE05)*, 2005.
13. M. Hepp and J. de Bruijn. Gen tax: A generic methodology for deriving OWL and RDF-S ontologies from hierarchical classifications, thesauri, and inconsistent taxonomies. In *ESWC*, 2007.
14. H. Knublauch, M. A. Musen, and A. L. Rector. Editing description logic ontologies with the Protégé OWL plugin. In *Description Logics*, 2004.
15. G. Miller. *WordNet: An electronic Lexical Database*. MIT Press, 1998.
16. H. S. Pinto, S. Staab, and C. Tempich. Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In *ECAI*, pages 393–397, 2004.
17. A. L. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, and C. Wroe. OWL pizzas: Practical experience of teaching OWL-DL: Common errors & common patterns. In *EKAW*, pages 63–81, 2004.
18. Dagobert Soergel, Boris Lauser, Anita Liang, Frehiwot Fisseha, Johannes Keizer, and Stephen Katz. Reengineering thesauri for new applications: The agrovoc example. *J. Digit. Inf.*, 4(4), 2004.
19. M. van Assem, A. Gangemi, and G. Schreiber. RDF/OWL representation of WordNet, W3C working draft, June 2006.
20. M. van Assem, M. R. Menken, G. Schreiber, J. Wielemaker, and B. J. Wielinga. A method for converting thesauri to RDF/OWL. In *International Semantic Web Conference*, pages 17–31, 2004.

16

21. I. Zaihrayeu, L. Sun, F. Giunchiglia, W. Pan, Q. Ju, M. Chi, and X. Huang. From web directories to ontologies: Natural language processing challenges. In *ISWC/ASWC*, pages 623–636, 2007.

Multilingual Ontology Mapping: Challenges and a Proposed Framework

(Extended Abstract)

Bo Fu[1], Rob Brennan[1] and Declan O'Sullivan[1]

Abstract. A key problem in supporting multilingual information retrieval and digital content management is reasoning about overlapping context domains. Ontologies are currently emerging as representation techniques for overlapping complimentary context domains. To date, research has focused on the mappings of monolingual ontologies, however, the issue of mapping ontologies written in different natural languages is relatively unexplored at the moment. This paper discusses challenges in the area of multilingual ontology mapping and proposes the semantic oriented mapping for multilingual ontologies (SOMMO) framework to advance the state of the art in multilingual ontology mapping. The SOMMO framework aims to improve multilingual ontology mapping results generated from existing monolingual ontology matching techniques by evaluating the semantics embedded in both the source and target ontologies.

1 INTRODUCTION

In recent years, ontologies have gained a large amount of attention as a part of the process of achieving semantic interoperability. Usage of ontologies traverses many disciplines, in Agriculture, the Agricultural Ontology Service[2] from the Food and Agriculture Organization (FAO) provides reference standardization for defining and structuring Agricultural terminologies. Since all FAO official documents must be made available in Arabic, English, Chinese, French and Spanish, a large amount of research has been carried out in the translations of large multilingual agricultural thesauri [1], mapping methodologies for them [2] [3] and a definition of requirements to improve the interoperability of these multilingual information resources [4]. In education, the Bologna declaration has introduced an ontology-based framework for qualification recognition [5] across the European Union, in an effort to best match labour markets with employment opportunities. In e-learning, educational ontologies are used to enhance learning experience [6], and to empower system platforms with high adaptivity [7]. In finance, ontologies are used to model knowledge in the stock market domain [8] and portfolio management [9]. In medicine, ontologies are employed to improve knowledge sharing and knowledge reuse, for example, a notable amount of research has focused on the creation of a traditional Chinese medicine ontology [10].

Usage of ontologies grew not only in terms of the number of application domains but also in their choices of natural languages as researchers across borders began to build domain specific knowledge bases. Reasoning and mapping of these multilingual ontologies thus has become a pressing issue. Given large and complex multilingual ontologies, it is unlikely that the mapping process would be practical if solely based on human processing, therefore, fully/semi-automated multilingual ontology mapping systems are needed.

The concept of creating ontologies that comprise different natural languages was explored when Carpuat et al [14] merged thesauri that were written in English and Chinese into one bilingual thesaurus in order to minimize repetitive work while building ontologies. A language-independent, corpus-based approach was employed to merge *WordNet*[3] - written in English, and *HowNet*[4] - written in Chinese by aligning synsets from the former and definitions of the latter. Similar research [15] has been done to match Dutch thesauri to *WordNet* by using a bilingual dictionary, and concluded a methodology for vocabulary alignment of thesauri written in different languages. Such methods succeed in aligning large numbers of words, however, they do not take structural aspects into account. Due to the nature of thesauri - being large collections of words, definitions and synonyms - ignoring their structures when generating a mapping poses little problem. Given the more complex structure and sophisticated class relationships of ontologies, such a method would be insufficient as the structures of these ontologies cannot be over-looked to form accurate mapping results.

Espinoza et al. [16] demonstrate a tool – *LabelTranslator* to empower end-users with choices of natural language when gaining knowledge from a given ontology, it is designed to ensure information represented in an ontology using one particular natural language would still achieve the same level of knowledge expressivity if translated into another natural language. The name, *LabelTranslator* is self-explanatory, it translates labels in a given ontology into one of three natural languages, English, Spanish and German. Users are allowed to select any label in a given ontology for *LabelTranslator* to translate, which then returns the selected term's translation along with its namespace and description. The system is comprised of seven steps: users must first tell the system which labels to translate; the system then translates the selected terms using lexical resources and translation web services, if compound

[1] Knowledge and Data Engineering Group, School of Computer Science and Statistics, Trinity College Dublin, Ireland. Email: {bofu, rob.brennan, declan.osullivan}@cs.tcd.ie.

[2] http://www.fao.org/aims/aos.jsp

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

[4] http://www.keenage.com

words are presented, they will be split into components for the translators; for each translated term, the system obtains a list of senses which is then used for disambiguation; in order to return the most appropriate translations, *LabelTranslator* determines the context by retrieving sets of other labels that are associated with the selected terms; it then lists senses for these labels in context; for each candidate translation, their senses are ranked by comparisons made to the context senses to produce a rank list; once the correct sense is selected from this ranked list, it finally updates the linguistic information of the ontology. In *LabelTranslator*, labels are selected one at a time by the user, the translation and description of the selected label are then presented. In the process of translating labels to a preferred natural language other than the original one, it aids the user to better understand the subject area. While *LabelTranslator* highlights challenges when translating multilingual ontologies automatically and includes sophisticated sense disambiguation mechanisms, it is built to translate an ontology from one natural language into another so that it is human readable, however, it does not deliver translated machine readable ontology documents so that software agents could manipulate and annotate.

Pazienza & Stellato [17] propose a linguistically motivated approach to ontology mapping, the framework urges the usage of linguistically enriched expressions when building an ontology and envisions systems that can automatically discover the embedded linguistic evidence and establish alignments that support users to produce sound ontology mapping documents. A three-step methodology is proposed where sets of ontologies with readily embedded linguistic resources are built at the ontology development stage and are fed to the automatic mapping system. A plug-in, *Ontoling* was also developed for the ontology editor *Protégé*[5] that enables users to browse linguistic resources provided by *WordNet* and *FreeDict*[6] during the ontology creation process. Though this methodology promises improved mapping results, the multilingual enriched ontologies demanded by the framework are hard to come by when such specifications are not currently included in the OWL standardization [18] effort.

A large amount of research has been done in the area of monolingual ontology mapping [11], however, the concept of multilingual ontology mapping is relatively new. Matching contests such as the Ontology Alignment Evaluation Initiative[7] (OAEI) encourage the progression of automated ontology matching tools and recognize the importance of addressing issues that are associated with multilingual ontologies. In the most recent OAEI competition, a test scenario involving the mapping of web site directories written in English and Japanese was defined [12]. Among thirteen contestants, four took part in this test scenario, however, only one matching tool was able to submit results [13] to the program.

The main challenges for (semi-)automated multilingual ontology mapping and a proposed framework is discussed in the following section.

2 CHALLENGES IN MULTILINGUAL ONTOLOGY MAPPING & THE SOMMO FRAMEWORK

In a scenario where automated mapping of two ontologies that are written in different natural languages is desired, one approach to achieve such a process is by translating one of them into the natural language that is used by the other ontology, e.g. using machine translation techniques, before applying monolingual ontology matching techniques. In such a multilingual ontology mapping approach, challenges are mainly found in the ontology translation phase and the monolingual ontology matching phase.

Being able to identify the most appropriate translation results of ontology concepts is crucial in the ontology translation phase. It is the author's opinion that these translated concepts will hugely impact on the quality of ontology mapping results generated by existing matching tools, since lexical matching techniques currently tend to dominate in the most successful matching tools [12]. Regardless of recent advances in the development of monolingual ontology matching tools, challenges remain in the generation of accurate matching results. Among ten challenges identified by Shvaiko & Euzenat [19] in the field of ontology matching, the discovery of background knowledge of a specific ontology is an important issue, most recent progress as discussed in [20] [21] attempts to resolve this critical matter.

To address the aforementioned challenges, the semantic oriented mapping for multilingual ontologies (SOMMO) framework is proposed. For each class, instance and property in a source ontology that is to be translated, a collection of translation candidates can be generated using existing machine translation tools such as the *GoogleTranslate* API[8] and the SDL *FreeTranslation* online translator[9]. Using lexicon dictionaries and based on the knowledge represented in the target ontology, a target lexicon database can be created which stores sets of synonyms for all the target concepts. In order to choose the most preferred translation results for each source concept, the target lexicon data-store is used to influence the translation selection algorithm. The source translation candidates are first compared against the sets of synonyms, if matches are found, for each matched target term and/or synonyms, their immediate surrounding terms, i.e. semantics – parent, child, sibling – are collected, and are ranked based on the similarity of their surrounding terms to that of the source terms. The highest ranked target term will be chosen as the most preferred translation result for the source term. If no matches are found when the candidates and synonyms are compared, or when surrounding term comparisons conclude no similarities, the translation selection is solely based on the semantic representations of the source term. In such a case, for each translation candidate, a set of interpretive keywords can be collected which describe the meanings of these candidates using a dictionary. These keywords can then be compared to the surrounding terms of the source term. Based on matches of these keywords, translation candidates can be ranked, with the highest ranked candidate being chosen as the most

[5] http://protege.standord.edu
[6] http://www.freedict.com
[7] http://oaei.ontologymatching.org
[8] http://code.google.com/p/google-api-translate-java
[9] http://www.freetranslation.com
[10] http://jena.sourceforge.net
[11] http://alignapi.gforge.inria.fr

preferred translation result. If no keywords match the source's surrounding terms, a translation result is generated by an automated machine translator. Using tools such as the Jena framework[10], the source structure can be rebuilt, together with the translated entities and expressions of the source concepts, a translated source ontology document can be created.

Given the target ontology and a translated source ontology – now both represented in the same natural language – matching relationships can be determined by applying existing monolingual ontology matching techniques such as the *Alignment* API[11]. Finally, the metadata gathered during the ontology translation phase are of use in the final monolingual matching process, since relationships were already established between some source terms and target terms when the latter are used to influence the translation outcomes of the former. Together with existing monolingual ontology matching tools, such metadata can assist the rendering of more accurate and higher confidence matching results between the translated source ontology and the target ontology. Hence reliable matching relationships are generated between concepts from the original source ontology and the target ontology regardless of their natural languages originally used.

The development of the SOMMO framework is part of on-going research work and several test cases are being designed to evaluate such an approach.

ACKNOWLEDGMENT

This research is partially funded by Science Foundation Ireland (SFI) as part of the National Development Plan (NDP) 2007-2013.

REFERENCES

[1] Chang C. and Lu W., The Translation of Agricultural Multilingual Thesaurus, in Proceedings of the 3rd Asian Conference for Information Technology in Agriculture, 2002.

[2] Liang A., Sini M., Chang C., Li S., Lu W., He C. and Keizer J., The Mapping Schema from Chinese Agricultural Thesaurus to AGROVOC, 6th Agricultural Ontology Service (AOS) Workshop on Ontologies: the more practical issues and experiences, 2005.

[3] Liang A. and Sini M., Mapping AGROVOC and the Chinese Agricultural Thesaurus: Definitions, tools, procedures, New Review of Hypermedia and Multimedia pp. 51 -- 62, 12 (1) 2006.

[4] Caracciolo C., Sini M. and Keizer J., Requirements for the Treatment of Multilinguality in Ontologies within FAO, Food and Agricultural Organisation of the United Nations, 2007.

[5] Vas R., Educational Ontology and Knowledge Testing, The Electronic Journal of Knowledge Management of Volume 5 Issue 1, pp. 123 -- 130, 2007.

[6] Cui G., Chen F., Chen, H. and Li S., OntoEdu: A Case Study of Ontology-based Education Grid System for E-learning, The Global Chinese Conference on Computers in Education conference, 2004.

[7] Sosnovsky S. and Gavrilova T., Development of Educational Ontology for C-programming, International Journal "Information Theories and Applications" Volume 13, pp. 303 – 308, 2006.

[8] Alonso L. S., Bas L. J., Bellido S., Contreras J., Benjamins R. and Gomez M. J., WP10: Case Study eBanking D10.7 Financial Ontology, Data, Information and Process Integration with Semantic Web Services, FP6-507483, 2005.

[9] Zhang Z., Zhang C. and Ong S. S., Building an Ontology for Financial Investment, in Intelligent Data Engineering and Automated Learning - IDEAL 2000, Data Mining, Financial Engineering, and Intelligent Agents, Second International Conference, pp. 308 – 313, 2000.

[10] Fang K., Chang C. and Chi Y., Leveraging Ontology-Based Traditional Chinese Medicine Knowledge System: Using Formal Concept Analysis, in Proceedings of the 9th Joint Conference on Information Sciences, 2006.

[11] Euzenat J. and Shvaiko P., Ontology Matching, Springer 2007.

[12] Multilingual Directory Data Set Specification, http://ri-www.nii.ac.jp/OAEI/2008, last accessed December 2008.

[13] Ontology Alignment Evaluation Initiative 2008 Results, http://oaei.ontologymatching.org/2008/results, last accessed December 2008.

[14] Carpuat M., Ngai G., Fung P. and Church W. K., Creating a Bilingual Ontology: A Corpus-Based Approach for Aligning WordNet and HowNet, In Proceedings of the 1st Global WordNet Conference, 2002.

[15] Malaise V., Isaac A., Gazendam L. and Brugman H., Anchoring Dutch Cultural Heritage Thesauri to WordNet: Two Case Studies, Proceedings of the Workshop on Language Technology for Cultural Heritage Data, 2007.

[16] Espinoza M., Gomez-Perez A. and Mena E., LabelTranslator – A Tool to Automatically Localize an Ontology, ESWC 2008, LNCS 5021, pp. 792 – 796, Springer 2008.

[17] Pazienza T. M. and Stellato A., Linguistically Motivated Ontology Mapping for the Semantic Web, Semantic Web Applications and Perspectives 2005, second Italian Semantic Web Workshop, 2005.

[18] OWL 2 Web Ontology Language Profile, http://www.w3.org/TR/2008/WD-owl2-profiles-20081202, last accessed December 2008.

[19] Shvaiko P. and Euzenat J., Ten Challenges For Ontology Matching, in Proceedings of the 7th International Conference on Ontologies, DataBases and Applications of Semantics (ODBASE) 2008.

[20] M. Sabou, M. d'Aquin and E. Motta, Exploring the Semantic Web as Background Knowledge for Ontology Matching, Journal on Data Semantics XI, LNCS Vol. 5383, pp. 156-190, Springer 2008.

[21] F. Giunchiglia, P. Shvaiko and M. Yatskevich, Semantic Matching, Encyclopedia of Database Systems, 2009, to appear.

Spatial Groundings for Meaningful Symbols

Stefan Dietze[1], **Vlad Tanasescu**[2]

Abstract. The increasing availability of ontologies raises the need to establish relationships and make inferences across heterogeneous knowledge models. The approach proposed and supported by knowledge representation standards consists in establishing formal symbolic descriptions of a conceptualisation, which, it has been argued, lack grounding and are not expressive enough to allow to identify relations across separate ontologies. Ontology mapping approaches address this issue by exploiting structural or linguistic similarities between symbolic entities, which is costly, error-prone, and in most cases lack cognitive soundness. We argue that knowledge representation paradigms should have a better support for similarity and propose two distinct approaches to achieve it. We first present a representational approach which allows to ground symbolic ontologies by using Conceptual Spaces (CS), allowing for automated computation of similarities between instances across ontologies. An alternative approach is presented, which considers symbolic entities as contextual interpretations of processes in spacetime or Differences. By becoming a process of interpretation, symbols acquire the same status as other processes in the world and can be described (tagged) as well, which allows the bottom-up production of meaning.

1 INTRODUCTION[2]

The widespread use of ontologies as a knowledge engineering device [15] together with the increasing availability of representations of overlapping domains of interest, raises the need to integrate distinct ontologies. This becomes crucial when considering the exploitation of the growing *Semantic Web (SW)* which naturally consists of multiple distributed ontological representations. Following a symbolic representation approach – as done by established representation standards such as RDF-S [29] and OWL [28] – requires the heterogeneity across distinct formalisations to be addressed and relationships between entities across ontologies to be (a) identified and (b) explicitly represented. Hence, formal relations are to be established between a set of knowledge entities E_1 from an ontology O_1 with the corresponding entities E_2 in a distinct ontology O_2 [7][26]. The expression *set of entities* here refers to the union of all concepts *C*, instances *I*, relations *R* and axioms *A* defined in a particular ontology. In that, the identification and representation of *similarities* [1] between entities across different ontologies, appears to be a necessary requirement to support interoperability between multiple heterogeneous ontologies.

However, with respect to this goal, several issues have to be taken into account. The symbolic approach proposed by knowledge representation and SW standards – describing symbols by using other symbols – has been criticised for lacking the grounding to a cognitive or perceptual level, what is known as the symbol grounding problem [16]. Without a grounding – i.e. linking symbols to cognition and to the observable reality - heterogeneity across ontologies cannot be handled appropriately [3][20]. Describing all aspects of a specific concept using symbolic representations is a costly task as well as a doubtful one, as the intended meaning of a symbolic concept usually depends on the context of its usage [25].

Due to these issues, in order to address (a), i.e. to identify knowledge entities which represent the same or similar meaning in distinct ontologies, current *ontology mapping* approaches have to exploit similarities at the symbolic level, e.g. based on linguistic or structural similarities across entities [8][13][19][7][26]. But such manual or semi-automatic identification of similarity relationships is also costly and prone to errors. Moreover, since knowledge entities across distinct ontologies usually represent real-world concepts which resemble each other just to a certain extent, representation of the gradual notion of similarity as in (b) is another challenge. Several approaches from the field of *fuzzy logic* aim at the representation of fuzzy and gradual relationships [2][13][23]. These approaches usually rely on the explicit, manual representation of relationships what is a costly and error-prone process as well, and also, tends to capture the subjective viewpoint of one individual.

Therefore, representational frameworks which enable to implicitly describe similarities across ontologies are required to fully facilitate ontology interoperability. Several approaches try to automate the computation of similarities through spatially oriented knowledge representation models. The *Conceptual Spaces (CS)* theory [11] proposes to describe concepts by gradual levels of abstraction starting with elementary sensory features, in order to bridge between the cognitive and the symbolic world. Concepts are represented as multidimensional *Vector Spaces (VS)*, and instances are represented as vectors, i.e. points, in these spaces. *Soft Ontologies (SO)* [17] follow a similar approach by representing a knowledge domain *D* through a multi-dimensional *ontospace A.* An item *I*, i.e. an instance, is represented by scaling each dimension to reflect its impact, presence or probability in the case of *I*. In that, a SO can be perceived as a CS where dimensions are measured exclusively on a ratio-scale. Hence, by relying on measurement-based representation of perceptual features, CS, VS and SO enable the automatic computation of instance similarity by means of distance metrics such as the Euclidean, Taxicab or Manhattan distance [18] or the Minkowsky Metric [24]. However, similarity computation requires the description of concepts through quantifiable metrics, even in case of qualitative characteristics. Moreover, these representation approaches do not provide the

[1] Knowledge Media Institute, The Open University, MK76AA, UK, Email: s.dietze@open.ac.uk.
[2] School of Arts, The University of Edinburgh, EH1 IJZ, UK, Email: vlad.tanasescu@ed.ac.uk.

means to represent arbitrary relations [22], such as *part-of* relations, common to symbolic knowledge models. In this regard, it is even more obstructive that the scope of a dimension cannot be defined, i.e. a dimension always applies to the entire CS/SO. For example the colour dimension applies to the whole entity rather than to parts of it [22]. Moreover, it can be argued, that representing an entire knowledge model through a coherent spatial representation, e.g. a CS, might not be feasible, particularly when attempting to maintain the meaningfulness of the spatial distance as a similarity measure.

Another issue with these approaches is that the spaces described are linked to cognitive structures, not to the environment itself. It could be argued that cognition mirrors the environment and that therefore such an approach is grounded. However, this grounding, as in the case of symbols, is only implicit. Moreover, when dimensions of a CS are linked to actual space and time, for example to represent the movement or growth of an entity, actual space and time have to be modeled as explicit conceptual spaces, detached from the environment. A more natural approach would be to consider spacetime as the underlying structure of all entities, and hence, of their conceptual representations, rather than a particular kind of "space".

2 ADDING MEANING TO SYMBOLS THROUGH SPATITEMPORAL GROUNDINGS

Spatiotemporal representations of knowledge are a promising approach to ontology grounding, even considering the previous issues. Indeed, space and time appear to be both cognitive and physical structures. Moreover distances seem the most natural way to represent similarity. We argue that some of the aforementioned issues can be alleviated by applying spatiotemporal structures to individual concepts instead of representing the whole ontology in a single spatial representation. We propose two approaches to achieve this.

2.1 Grounding Ontologies in Conceptual Spaces

A hybrid representational approach – combining symbolic ontologies with corresponding spatial representations – has the potential to enable similarity computation across ontologies. In that, we consider the representation of a set of *n* concepts *C* of an ontology *O* through a set of *n* spatial representations *SR*, where SR would be realized e.g. through a representation in a CS as proposed in previous work [5][22]. Figure 1 illustrates the grounding of ontologies in multiple CS as proposed in [5].

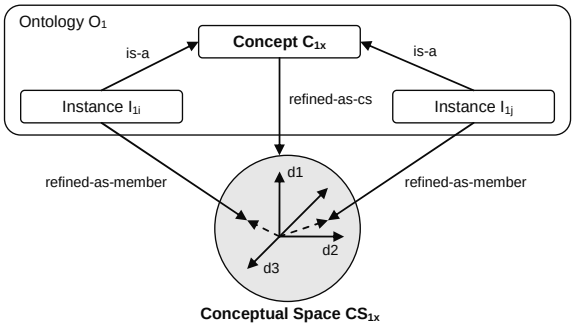


Figure 1. Grounding ontologies in multiple CS.

Note, in order to facilitate this vision, [5] proposes an ontology (CSO) which allows to refine any arbitrary concept as a CS instance while ontology instances are represented as member instances in CSO. After these additional steps, similarity between instances across distinct ontologies is computed by means of their Euclidean distance.

In [4][6] applications are proposed which make use of CSO and the representational approach described here to enable Semantic Web Service (SWS) [9] interoperability. Symbolic representations of the contexts which are either targeted by available SWS or desired by particular service consumers are refined by means of CSO. Based on similarity-computation within CS, the most similar SWS for a given query is being discovered and executed. In that, [4][6] prove the applicability of the proposed representational approach to contribute to the ontology alignment problem and provide detailed case studies from two distinct domains, eLearning and location-based applications.

While still benefiting from implicit similarity information, such a hybrid approach allows maintaining the advantages of ontological knowledge representations. As proposed in [5], spatial representations can be defined in dedicated ontologies. Such a two-fold representational approach allows the implicit representation of similarities between instances across heterogeneous ontologies, and consequently, provides a means to facilitate ontology interoperability. As shown in [4], applying this approach has the potential to reduce the effort required to align distinct heterogeneous ontologies and the extent to which two distinct parties have to share their conceptualisations. Whereas traditional ontology mapping methodologies rely on mechanisms to semi-automatically detect and formally represent similarities at the concept and the instance level, our approach just requires a common agreement at the concept level since similarity information at the instance level is implicitly defined.

2.2 Grounding Ontologies in Processual Spacetime

Another approach to the spatiotemporal grounding of ontologies, introduced in [25][26] and [27], considers reality as a processual continuum structured by spacetime. So-called objects are processes persisting in their form of function and only superficially detached from the larger processual flux. The symbolic approach of naming an element of the world is a process that isolates an entity according to a context. It is the variety of contexts (cultures, languages, purposes, etc.) which produces heterogeneity across ontologies. When isolated from the processual flux, but not yet integrated to a KR paradigm, such as CS or taxonomies, a meaningful entity can be called a *difference*. Differences represent processes and the regions of spacetime that they shape through their activity. They do not require a pre-existing formal conceptualisation, and can therefore appropriately be represented by *tags*. Tags, as everything else, are a part of the processual environment and can also be described, i.e. tagged. We have designed Tagopedia[3] to collect a user's tags and to allow the tag owner as well as other users to *tag the tags* themselves. For example *tank* can be tagged by user *u1* with *fish*, *u2* can tag it with *weapon* and *war*, and *u3* with *container*, and *u4* with *vehicle*. This extension of

[3] http://tagopedia.info/

collaborative tagging systems has been dubbed *extreme tagging systems*, or ETS. After tagging a tag, the user selects the type of relation between the two tags from a small set. One of the possible relations is *similarity* (e.g. between *tank* and *container* or *vehicle*), another is *copresence* which expresses the fact that two differences (the entities represented by the tags) are often found together in space and time (e.g. *tank*, *weapon* and *war* at the time period when the tagging occurs). Rather than considering the actual shape of a dimension like in CS or having to specify a ratio for characteristics like in SO, the resulting network is arranged according to similarity and frequence relations. A frequentist interpretation of probability provides weights to the graph links: the more often a relation is tagged as similar, the closer the node's meanings are, which shapes the corresponding space. This network can then be consulted in order to map entities from an ontology: the concept *tank* with *wheels* will be associated with *vehicle* from a target ontology even if this concept is not present in the source ontology. Other inferences are possible with this framework. For example, from the fact that *cat* is marked as *copresent* with *house*, one could infer, with the appropriate ontology, that a cat is a kind of pet.

Extreme tagging has been used in [26] to provide an emergent notion of place by linking ETS with Wordnet: tags recognized as geographical entities which were linked to differences recognized as affordances where identified as a geographical place relevant to the query. In [25] an ETS was used to discover relevant services at an appropriate scale. For example (cf. Figure 2), user *u1* is looking for emergency information about a town of interest. Town co-occurs with street in the system's ETS graph but street has not yet been geotagged in the same area. However, it has been linked to the service *s-roads* by user *u2*. This service is geotagged in that region at the appropriate scale and is therefore displayed.

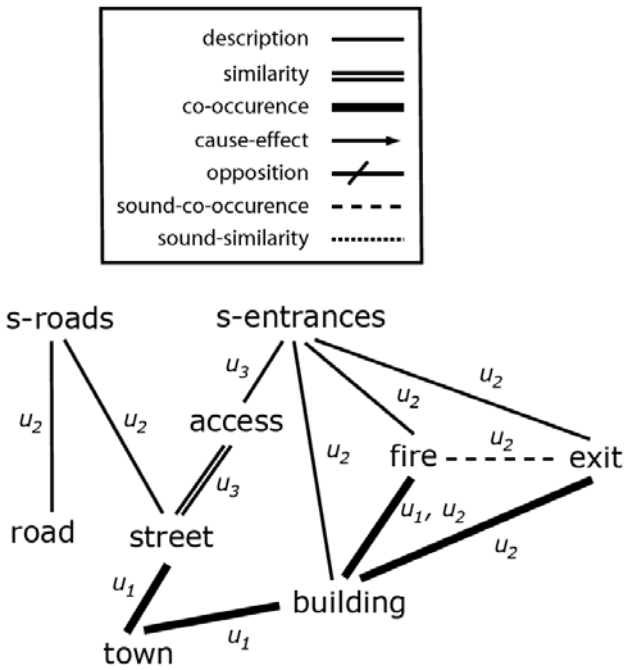


Figure 2. List of associations and ETS graph for service discovery.

In the *Cultures of Legibility* project[4] ETS will be used to present an image of the city defined by places involved in the daily rhythms of the inhabitants. By collecting information about the routines of city dwellers, as well as data regarding their geographical trace and tag based descriptions of the places of interest along these routes, the project aims to provide an image of the city formed by its usage rather than land use categories.

3 CONCLUSION

In this paper we presented two novel approaches which aim at alleviating the lack of of grounding of symbolic ontologies in order to ease the integration of heterogeneous knowledge models.

The first approach proposes a grounding of ontologies in spatial representations – such as CS – and allows for automated computation of similarities of instances across heterogeneous ontologies by means of their spatial distances in the set of shared CS. Hence, it extends symbolic ontologies with grounding to a cognitive level and hence, facilitates similarity computation across ontologies while still taking advantage of the knowledge represented at the symbolic level, such as arbitrary relations between knowledge entities.

The second approach, based on grounding in processual spacetime, offers a bottom-up method to the production of meaning. Similarity is defined by users according to various contexts, which ensures that the result is cognitively sound. Co-occurence ensures a link with the processual environment, and therefore a grounding in reality.

Nevertheless, the contributions stated above come to a certain cost. The first approach (Section 2.1) requires additional effort to establish spatial groundings based on measurements and some issues related with VS-based knowledge representation still remain. For instance, whereas defining instances, i.e. vectors, within a given VS appears to be a straightforward process of assigning specific quantitative values to quality dimensions, the definition of the VS itself is not trivial at all and dependent on individual perspectives and subjective appraisals. The second approach (Section 2.2) leads to issues related to appealing to the "wisdom of crowds" which can be biased or inappropriate for some domains. However, the possibility to restrict the resulting network to the descriptions of members of selected communities can alleviate this. VS-based approaches appear to not fully solve the symbol grounding issue but to shift it from the process of describing instances to the definition of the spatial representation, and the need may occur to align the spaces themselves. Therefore future work on the links of these with actual spatiotemporal processes is needed. Nevertheless, as instance similarity computation becomes an increasingly important challenge, the further investigation of spatial groundings for symbolic representation models seems to us an essential step towards the vision of interoperable ontologies.

[4] http://ddm.caad.ed.ac.uk/groups/jakarta/

REFERENCES

[1] Bisson, G. (1995). Why and how to define a similarity measure for object based representation systems. Towards Very Large Knowledge Bases, pages 236–246, 1995.

[2] Calegari, S., Ciucci, D.: Integrating Fuzzy Logic in Ontologies. In: Manolopoulos, Y., Filipe, J., Constantopoulos, P., Cordeiro, J. (eds.) ICEIS, pp. 66–73. INSTICC press (2006)

[3] Cregan, A. (2007), Symbol Grounding for the Semantic Web. 4th European Semantic Web Conference 2007, Innsbruck, Austria.

[4] Dietze, S., Gugliotta, A., Domingue, J., (2008) Conceptual Situation Spaces for Situation-Driven Processes. 5th European Semantic Web Conference (ESWC), Tenerife, Spain.

[5] Dietze, S., and Domingue, J. (2008) Exploiting Conceptual Spaces for Ontology Integration, Workshop: Data Integration through Semantic Technology (DIST2008), Workshop at 3rd Asian Semantic Web Conference (ASWC) 2008, Bangkok, Thailand, URL: http://events.sti2.at/dist2008/papers/ExploitingConceptualSpacesForOntologyIntegration-dist2008.pdf.

[6] Dietze, S., Gugliotta, A., Domingue, J., (2008) Bridging the Gap between Mobile Application Contexts and Semantic Web Resources. Chapter in: Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability: Adaptive Technologies and Applications, Editor: Dragan Stojanovic, Information Science Publishing (IGI Global), November 2008.

[7] Ehrig, M, Sure, Y. (2004), Ontology Mapping - An Integrated Approach, in Proceedings of ESWS, 2004.

[8] Euzenat, J., Guegan, P., and Valtchev, P. OLA in the OAEI 2005 Alignment Contest. K-Cap 2005 Workshop on Integrating Ontologies 2005, 97-102.

[9] Fensel, D., Lausen, H., Polleres, A., de Bruijn, J., Stollberg, M., Roman, D., Domingue, J. (2006): Enabling Semantic Web Services – The Web service Modelling Ontology, Springer 2006.

[10] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L. (2002). Sweetening Ontologies with DOLCE. In: A. Gómez-Pérez , V. Richard Benjamins (Eds.) Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web: 13th International Conference, EKAW 2002, Siguenza, Spain, October 1-4, 2002.

[11] Gärdenfors, P. (2000), Conceptual Spaces - The Geometry of Thought. MIT Press, 2000.

[12] Gärdenfors, P. (2004), How to make the semantic web more semantic. In A.C. Vieu and L. Varzi, editors, Formal Ontology in Information Systems, pages 19–36. IOS Press, 2004.

[13] Gabora, L., Rosch, E. and Aerts, D. (2008) Toward an Ecological Theory of Concepts, Ecological Psychology, 20, pp. 84-116

[14] Gao, M., Liu, C., 2005. Extending OWL by Fuzzy Description Logic, Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05).

[15] Guarino, N. (1995), Formal Ontology, Conceptual Analysis and Knowledge Representation. International Journal of Human-Computer Studies, Vol. 43, Issues 4-5, pp. 625-640.

[16] Harnad, S. (1990) The symbol grounding problem, Physica D: Nonlinear Phenomena, 42, pp. 335-346

[17] Kaipainen, M., Normak, P., Niglas, K., Kippar, J., Laanpere, M., Soft Ontologies, spatial Representations and multi-perspective Explorability, Expert Systems, November 2008, Vol. 25, No. 5.

[18] Krause, E. F. (1987). Taxicab Geometry. Dover.

[19] Moncan, A., Cimpian, E., Kerrigan, M., Formal Model for Ontology Mapping Creation, in I. Cruz et al. (Eds.): ISWC 2006, LNCS 4273, pp. 459–472, 2006. Springer-Verlag Berlin Heidelberg 2006.

[20] Nosofsky, R. (1992), Similarity, scaling and cognitive process models, Annual Review of Psychology 43, pp. 25- 53, (1992).

[21] Pease, A., Niles, I., Li, J.(2002), The suggested upper merged ontology: A large ontology for the semanticweb and its applications. In: AAAI-2002Workshop on Ontologies and the Semantic Web. Working Notes (2002)

[22] Schwering, A. (2005). Hybrid Model for Semantic Similarity Measurement, in R. Meersman and Z. Tari (Eds.): CoopIS/DOA/ODBASE 2005, LNCS 3761, pp. 1449 – 1465, 2005.

[23] Straccia, U.: A fuzzy description logic for the semantic web. In: Sanchez, E. (ed.) Fuzzy Logic and the Semantic Web. Capturing Intelligence, pp. 73–90. Elsevier, Amsterdam (2006)

[24] Suppes, P., D. M. Krantz, et al. (1989). Foundations of Measurement - Geometrical, Threshold, and Probabilistic Representations. San Diego, California, USA, Academic Press, Inc.

[25] Tanasescu, V. and Streibel, O. (2007) Chen, L.; Cudré-Mauroux, P.; Haase, P.; Hotho, A. & Ong, E. (ed.) Extreme Tagging: Emergent Semantics through the Tagging of Tags ESOE, CEUR-WS.org, 292, pp. 84-94

[26] Tanasescu, V. and Domingue, (2008) J. B. Boll, S.; Jones, C.; Kansa, E.; Kishor, P.; Naaman, M.; Purves, R.; Scharl, A. & Wilde, E. (ed.) A Differential Notion of Place for Local Search LocWeb, ACM, 300, pp. 9-16.

[27] Tanasescu, V., Roman, D. and Domingue, J.B.. (2009). Service Selection via Extreme Geotagging, in proceedings of the International Conference on Advanced Geographic Information Systems & Web Services (GEOWS), 2009.

[28] W3C Web Ontology Language: http://www.w3.org/OWL/

[29] W3C RDF Schema: http://www.w3.org/TR/rdf-schema/

[30] Xiaomeng Su. (2002). A text categorization perspective for ontology mapping, Technical report, Department of Computer and Information Science, Norwegian University of Science and Technology, Norway, 2002.

Ontology Correspondence via Theory Interpretation

Immanuel Normann[1] and **Oliver Kutz**[2]

Abstract. We report on ongoing work to apply techniques of automated theory morphism search in first-order logic to ontology matching and alignment problems. Such techniques are able to discover 'structural similarities' across different ontologies by providing theory interpretations of one ontology into another.

We sketch the techniques currently available for automating the task of finding theory interpretations in first-order logic and discuss possible extensions and modifications for other ontology languages such as description logics and modular ontology languages such as $\mathcal{E}$-connections.

1 Introduction and Motivation

The problem of finding semantically well-founded correspondences between ontologies, possibly formulated in different logical languages, is a pressing and challenging problem. Ontologies may be about the same domain of interest, but may use different terms; one ontology might go into greater detail than another, or they might be formulated in different logics, whilst mostly formalising the same conceptualisation of a domain, etc. To allow re-use of existing ontologies and to find overlapping 'content', we need means of identifying these 'overlapping parts'.

Often, ontologies are mediated on an ad-hoc basis. Clearly, any approach relying exclusively on lexical heuristics or manual alignment is too error prone and unreliable, or does not scale. As noted for instance by [16], even if a first matching is realised automatically using heuristics, a manual revision of such candidate alignments is still rather difficult as the semantics of the ontologies generally interacts with the semantics given to alignment mappings.

A lot of research has already been carried out in the area of ontology matching [6]. However, most work is based on approximate matching of the graph structures of taxonomies and statistical or heuristic approaches, see e.g. [10, 9].

A new approach, that we currently explore, is to apply methods of automated theory interpretation search to the realm of ontologies. Such methods have been mainly developed for the application to formalised mathematics (and some of the techniques currently are specialised for theories formulated in first-order logic). Whilst theory interpretations are rather flexible in that they are not restricted to exact formulation and phrasing of ontology terms, in contrast to the above mentioned approaches to ontology matching, they do establish a logically rather strict relationship across two ontologies, namely that all axioms of one ontology are provable in the other along a translation, essentially embedding one ontology into another.

Such embeddings can give guidance in ontology development, and can be applied for searching and structuring of 'design patterns' for ontologies.

[1] Department of Linguistics and Literature, University of Bremen, Germany, email: normann@uni-bremen.de

[2] SFB/TR 8 Spatial Cognition, University of Bremen, Germany, email: okutz@informatik.uni-bremen.de

2 Theory Interpretations and Refinements

Theory interpretations have a long history in mathematics generally, and are probably employed by any 'working mathematician' on a daily basis; the basic idea is the following: given two theories T_1 and T_2 (which we here assume to be first-order theories), find a mapping of terms of T_1 to terms of T_2 (a signature morphism, typically expected to respect typing) such that all translations of axioms of T_1 become provable from T_2. If such a theory interpretation is successfully provided, all the knowledge that has already been collected w.r.t. T_1 can be re-used from the perspective of T_2, using the translation (see [7] for some examples from the history of mathematics). In this case, in mathematical jargon, we might say that T_2 **carries the structure** of T_1.

Certain, very basic structures, are found everywhere in mathematics. The most obvious example might be group theory. The basic abstract structure of a group can be re-interpreted in a more concrete setting, giving the group in question additional structure (think of the natural numbers, rings, vector-spaces, etc.). Re-using the metaphor mentioned above, we say that an ontology O_2 *carries the structure of* O_1, if the latter can be re-interpreted, by an appropriate translation σ, into the language of O_2 such that all of its axioms are entailed by O_2. In this case, informally, we consider the pair $\langle O_2, \sigma \rangle$ a **context** for O_1.

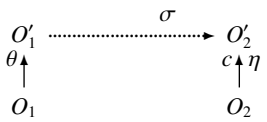


Figure 1. A heterogeneous refinement/theory interpretation.

The notion of theory interpretation is also closely related to the notion of refinement from software engineering. A heterogeneous refinement is depicted in Fig. 1. Here, given ontologies O_1 and O_2, possibly formulated in different logics, say a DL and a variant of first-order logic, we want to show that O_2 specialises, or refines, the information contained in O_1. To do this, we first need to translate

both O_1 and O_2 into a common logic, say first-order logic, by means of suitable translations θ and η. Here, the translation η additionally needs to be conservative in order not to 'distort' the information contained in O_2. In a final step, a theory interpretation σ from O_1' to O_2' is provided, showing that all translations of axioms of O_1 hold in O_2' along $\theta \circ \sigma$. The notion of a heterogeneous refinement also leads to a general definition of heterogeneous sub-ontology, compare [13].

It should be clear that whenever either the logics or the signatures of the ontologies involved do not directly fit, there are a number of possible solutions to choose from (we can just extend the logic in question, we can extend definitionally the signature, or both).[3]

Here is an illustrative example from mathematics:

Example 1 (Lattices and Partial Orders) *Consider P as the theory of partial-orders with* $\mathbf{Sig}(P) = \{\leq\}$ *and let L be the theory of lattices with* $\mathbf{Sig}(L) = \{\sqcap, \sqcup\}$. *These are both first-order theories, so the logical languages are directly compatible and we only need to translate the non-logical terms. However, the signatures obviously do not fit as L has only binary functions (rather than relations). This can be remedied by extending the signature of L by a binary relation symbol $\sqsubseteq$ (which makes the signatures fit by the mapping $\sigma : \leq \mapsto \sqsubseteq$), and define $\mathfrak{S} = \{\forall a, b. a \sqsubseteq b \leftrightarrow a \sqcup b = a\}$. This is a definitional axiom. It can now be seen that $L \cup \mathfrak{S} \models \sigma(P)$, i.e. σ is a theory interpretation embedding the theory of partial orders into the theory of lattices, using the definitional axiom in $\mathfrak{S}$.*

Thus, we may say that lattices carry the structure of partial orders. It should be obvious that both these theories also define central structures for ontology design.

3 Automated Discovery of Theory Interpretations

The goal of discovering ontology interpretations may be rephrased as the problem of finding all those ontologies in a large repository $\mathfrak{R}$ that could serve as a context (in the above sense) for a given ontology O_1. I.e. given O_1, we are looking for the set

$$\{O_2 \in \mathfrak{R} \quad | \quad O_1 \text{ is interpretable into } O_2\}.$$

Conversely, given O_2, we can look for the set

$$\{O_1 \in \mathfrak{R} \quad | \quad O_2 \text{ is interpretable into } O_1\},$$

i.e. the set of all ontologies into which O_2 can be interpreted.

In case of ontologies formalised in **FOL**, this task is undecidable, whereas for ontologies formalised in DL it is generally decidable. I.e., given the ontologies O_1, O_2, and a signature morphism σ from O_1 to O_2, it is decidable whether the σ-translated axioms of O_1 are entailed by O_2. However, the combinatorial explosion yielded by trying to find all possible symbol mappings between two given ontologies makes such a brute force approach unpractical.

To obtain one of the answer sets above in reasonable time (i.e seconds or minutes), we necessarily have to relax our initial goal towards an approximation of the set of all possible contexts for a given ontology. In summary, our approach for the first-order caseis based on formula matching modulo an equational theory—elaborated in detail in [20]. We want to outline this in the following.

Suppose we are given a source ontology O_1 and a target ontology O_2, which we assume have been translated to first-order via the standard translations. In the first step, we normalise each sentence of these ontologies according to a fixed equational theory. The underlying technique basically stems from term-rewriting: rewrite rules represent an equational theory such that all sentence transformations obtained through these rules are in fact equivalence transformations, e.g. such as $\neg A \sqcap \neg B \mapsto \neg(A \sqcup B)$. A normal form of a convergent rewrite system is then the unique representative of a whole equivalence class of sentences. The goal of normalisation is thus to identify (equivalent) expressions such as $\neg(\exists R.A \sqcap B)$ and $\neg B \sqcup \forall R.\neg A$.

In the next step, we try to translate each normalised axiom φ from O_1 into O_2, i.e. we seek a sentence ψ in O_2 and a translation σ such that $\sigma(\varphi) = \psi$. Note that potentially each axiom can be translated to several target sentences via different signature morphisms. To translate all axioms of O_1 into O_2, there must be a combination of *compatible* signature morphisms[4] determined from the previous, single sentence matchings. This task is also known as (consistent) many-to-many formula matching. In fact many-to-many formula matching modulo some equational theory is already applied in automated theorem proving (ATP) [11]. However, our approach is different in a crucial aspect: it allows for significant search speed up. We are normalising all ontologies as soon as they are inserted into the repository, i.e. not at cost of query time. Only the normalisation of the query ontology is at query time. Moreover, the normal forms not just allow for matching modulo some equational theory, but also enable a very efficient matching pre-filter based on skeleton comparison. A sentence skeleton is an expression where all (non-logical) symbols are replaced by placeholders. E.g., $\square \sqsubseteq \square \sqcup \square$ is the skeleton of $A \sqsubseteq B \sqcup C$. Obviously, two sentences can only match if they have an identical skeleton. Since syntactic identity can be checked in constant time, a skeleton comparison is a very efficient pre-filter for sentence matching.

Concerning sentence normalisation, some further improvements in comparison to traditional normalisation in ATP should be mentioned. In ATP, formulae are typically normalised to CNF for resolution, or DNF for tableaux reasoning. Both are not unique normal forms (even not modulo associativity and commutativity (AC)). Our approach uses a Boolean ring normal form which is unique modulo AC. Moreover, we developed an AC standardisation that computes a unique skeleton for given AC-equivalence classes of sentences.

All the presented techniques were developed in the context of formalised mathematics and a tool for the automated discovery of theory interpretations in first-order logic has already been implemented [20]. This has been used for experiments on a **FOL** version of the Mizar library [18] that contains about 4.5 million formulae distributed in more than 45.000 theories, and thus is the world's largest corpus of formalised mathematics. Experiments where each theory was used as source theory for theory interpretation search in the rest of the library demonstrated the scalability of our approach. On average, a theory interpretation search takes about one second and yields 60 theory interpretations per source theory.

4 Discussion and Outlook

Because of the encouraging results in formalised mathematics, we are currently adopting and modifying these techniques for the application in the realm of ontologies. In principle, the methods for automated discovery of theory interpretations developed in [20] can

[3] E.g. the OneOf constructor found in many description logics allowing a finite enumeration of the elements of a concept is also expressible as a disjunction of nominals, and conversely. Such translations/simulations can be handled by a library of logic translations.

[4] Two signature morphisms are compatible if they translate all their common symbols equally.

be applied to any formalised content as long as the entailment relation obeys certain properties (as specified e.g. in entailment systems [17]).

Of course, there is no guarantee that what is successful for mathematical theories is equally successful for formal ontologies, and some of the characteristics and features regularly found in ontologies are problematic.

A central difference between formalised mathematics and ontologies is in the expressivity of the underlying formal languages: obviously, **FOL** (mostly used for formalised mathematics) is more expressive than typical DLs (used for ontologies). This is also reflected in the more complex grammar of **FOL**: DL typically completely lacks variables, often has no function symbols, and also no relations of arity greater than two. Hence, **FOL** formulae containing such constructs do not have a directly corresponding syntactical expression in DL. Intuitively, we may say that, compared to DL, there is a larger syntactic variety of **FOL** formulae. In practice, the majority of ontologies that can be found on the internet even make use of only a rather small fragment of the DL expressivity—for instance, ontologies which are just taxonomies have no other axioms than is-a hierarchies.

This difference in syntactic complexity between **FOL** and DL has most likely in many cases two (mutually dependent) unfavourable consequences for ontology morphism search: 1) a less effective skeleton filter and 2) lots of meaningless search results. Due to the lower structural variety of axioms in ontologies, many DL axioms share identical skeletons. Thus, on average, a given skeleton in DL does not reduce the search space for matching formulae in an ontology on the same scale as a skeleton in a **FOL** theory would. For the same reason, the chance to match a source formula to many target formulae is higher in DL ontologies than in **FOL** theories. In other words: it is generally likely that the number of interpretations between DL theories is much higher than between **FOL** theories. In many (if not most) cases, these DL interpretations may turn out to be meaningless, though. A typical example is an interpretation between taxonomies: if we consider the is-a hierarchy of a taxonomy as a tree, then ontology matching becomes essentially tree matching. Clearly, a 'small' tree can often be mapped into a 'large' tree in several ways. Since such a mapping does not at all depend on the node names of the involved trees (i.e. the terms of the ontologies), this means that there may be quite a few interpretations between taxonomies of completely unrelated domains. Such interpretations, however, are meaningless from a common sense perspective.

Initial experiments on DL ontologies already suggested some ideas on how to overcome these problems in future work:

- Interactive search space reduction: the user should be able to enforce some mappings of non-logical symbols—often some mappings are explicitly intended.
- Exploitation of the decidability of DLs for the morphism search.
- Specialised normal forms designed particularly for various DLs.

Many approaches to connecting, aligning, or linking ontologies, or to interpret the vocabulary (and thus re-use its axiomatisation) of one ontology in another, rely on notions of symbol mapping that are more complex than simple signature morphisms. Examples of such formalisms, which introduce additional semantic complexity, are distributed DLs [16, 3, 2] and $\mathcal{E}$-connections [15, 5]. The general semantic idea of these approaches is similar, and is illustrated in Fig. 2.

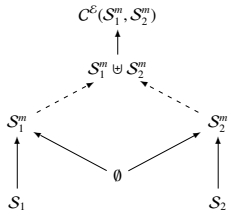


Figure 2. $\mathcal{E}$-connections or DDLs as structured heterogeneous theories

Here, given two ontologies $\mathcal{S}_1$ and $\mathcal{S}_2$, we first construct their disjoint union keeping the vocabulary completely disjoint. Given a 'link language' that allows to axiomatically connect the sorts of the component ontologies, we can in a second step provide a theory extension $C^{\mathcal{E}}(\mathcal{S}_1^m, \mathcal{S}_2^m)$, see [13] for technical detail. The nature of the 'link language' is here left open intentionally, as this is the main point of divergence between DDLs, $\mathcal{E}$-connections, and similar approaches.

The necessity of using such kinds of more expressive link or mapping languages has been shown in many application scenarios. [12], for instance, analyse the problem of relating an ontology encoding the linguistic spatial semantics of natural language utterances as represented in GUM [1] with spatial calculi, using the example of the double-cross calculus DCC [8] for projective relations (orientations).

Clearly, the problem of theory interpretation search takes a different turn in such a situation. Given ontologies $\mathcal{S}_1$ and $\mathcal{S}_2$, find an appropriate bridge theory $\mathcal{B}$ (for instance a set of bridge rules in the sense of [3]) and a signature morphism σ such that for all formulae ϕ (in an appropriate signature)

$$\mathcal{S}_1 \models \phi \text{ implies } \langle \mathcal{S}_1, \mathcal{S}_2, \mathcal{B} \rangle \models \sigma(\phi)$$

Whilst the bridge theory typically interacts with the semantics of $\mathcal{S}_1$ and $\mathcal{S}_2$, it is often natural to assume that $\mathcal{B}$ is conservative in at least one direction (see e.g. [4]). Different variants of this definition need to be analysed. Moreover, the algebraic equational theory that is used to identify equivalent formulae needs to be adapted in order to allow the identification of axioms loosely associated through a bridge theory.

Concerning automated reasoning support, a tool for the automated discovery of theory interpretations in first-order logic has already been implemented [20], and is currently being integrated into the Hets system [14, 19] with the aim of adding specialised routines for decidable ontology languages and corresponding integration problems. At the moment we perform experimental tests on a set of ontologies to evaluate the potential of first-order based theory interpretation search.

Acknowledgements

For the work reported in this article we gratefully acknowledge the financial support of the European Commission through the OASIS project (Open Architecture for Accessible Services Integration and Standardisation) and the Deutsche Forschungsgemeinschaft through the Collaborative Research Center on Spatial Cognition (SFB/TR 8). The authors would like to thank Joana Hois for fruitful discussions.

REFERENCES

[1] J. Bateman, T. Tenbrink, and S. Farrar, 'The Role of Conceptual and Linguistic Ontologies in Discourse', *Discourse Processes*, **44**(3), 175–213, (2007).

[2] A. Borgida, 'On Importing Knowledge from DL Ontologies: some Intuitions and Problems', in *Proc. of DL*, (2007).
[3] A. Borgida and L. Serafini, 'Distributed Description Logics: Assimilating Information from Peer Sources', *Journal of Data Semantics*, **1**, 153–184, (2003).
[4] B. Cuenca-Grau and O. Kutz, 'Modular Ontology Languages Revisited', in *Proc. of the IJCAI'07 Workshop on Semantic Web for Collaborative Knowledge Acquisition (SWeCKa), Hyderabad, India, January 2007*, (2007).
[5] B. Cuenca Grau, B. Parsia, and E. Sirin, 'Ontology Integration Using $\mathcal{E}$-Connections', in *Ontology Modularization*, eds., H. Stuckenschmidt and S. Spaccapietra, Springer, (2009). To Appear.
[6] J. Euzenat and P. Shvaiko, *Ontology Matching*, Springer, Heidelberg, 2007.
[7] W. M. Farmer, 'Theory Interpretation in Simple Type Theory', in *Higher-Order Algebra, Logic, and Term Rewriting*, volume 816 of *LNCS*, pp. 96–123. Springer, (1994).
[8] C. Freksa, 'Using orientation information for qualitative spatial reasoning', in *Theories and methods of spatio-temporal reasoning in geographic space*, volume 639 of *LNCS*, 162–178, Springer, (1992).
[9] F. Giunchiglia, F. Mcneill, M. Yatskevich, J. Pane, P. Besana, and P. Shvaiko, 'Approximate structure-preserving semantic matching', in *Proceedings of ODBASE*, 1217–1234, (2008).
[10] F. Giunchiglia, M. Yatskevich, and P. Shvaiko, 'Semantic Matching: Algorithms and implementation', *Journal on Data Semantics*, **IX**, 1–38, (2007).
[11] P. Graf, *Term Indexing*, volume 1053 of *Lecture Notes in Computer Science*, Springer, 1996.
[12] J. Hois and O. Kutz, 'Natural Language meets Spatial Calculi', in *Spatial Cognition VI. Learning, Reasoning, and Talking about Space. 6th International Conference on Spatial Cognition*, eds., C. Freksa, N. S. Newcombe, P. Gärdenfors, and S. Wölfl, LNCS, pp. 266–282. Springer, (2008).
[13] O. Kutz, D. Lücke, and T. Mossakowski, 'Heterogeneously Structured Ontologies—Integration, Connection, and Refinement', in *Advances in Ontologies. Knowledge Representation Ontology Workshop (KROW 2008)*, eds., T. Meyer and M. A. Orgun, volume 90 of *CRPIT*, pp. 41–50, Sydney, Australia, (2008). ACS.
[14] O. Kutz, D. Lücke, T. Mossakowski, and I. Normann, 'The $\mathcal{OWL}$ in the CASL—Designing Ontologies Across Logics', in *OWL: Experiences and Directions, 5th International Workshop (OWLED-08)*, October 26–27, ISWC, Karlsruhe, Germany, (2008).
[15] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev, '$\mathcal{E}$-Connections of Abstract Description Systems', *Artificial Intelligence*, **156**(1), 1–73, (2004).
[16] C. Meilicke, H. Stuckenschmidt, and A. Tamilin, 'Reasoning Support for Mapping Revision', *Journal of Logic and Computation*, (2008).
[17] J. Meseguer, 'General logics', in *Logic Colloquium 87*, pp. 275—329. North Holland, (1989).
[18] Mizar mathematical library. Web Page at `http://www.mizar.org/library`.
[19] T. Mossakowski, C. Maeder, and K. Lüttich, 'The Heterogeneous Tool Set', in *TACAS 2007*, eds., Orna Grumberg and Michael Huth, volume 4424 of *LNCS*, pp. 519–522. Springer, (2007).
[20] I. Normann, *Automated Theory Interpretation*, Ph.D. dissertation, Department of Computer Science, Jacobs University, Bremen, 2009.

Ontology Evolution through Agent Collaboration

Heather S. Packer and **Nicholas Gibbins** and **Nicholas R. Jennings** [1]

Abstract. We present a technique that enables a software agent to augment its ontology with domain related concepts by collaborating with other agents. The collaborating agents have their own individual ontologies, they can share concepts and relationships that relate to a requested specific concept (which is known as a fragment). Thus, specifically, our technique selects the fragments that will be shared. This approach enables agents to answer queries with more range and detail, and it also enables an agent to infer new exploitable knowledge. Without this capability, an agent may be limited by its domain model, and cannot reflect changes in the environment. Through empirical evaluation, we show that our technique reduces the cost of acquiring concepts that are regularly used (compared with learning nothing) and reduces the complexity of the agent's ontology by augmenting it with selected concepts and relationships which are related to its domain (compared with learning everything).

1 Introduction

Agents that model a domain for the purpose of answering queries can be limited to the knowledge instantiated in their model. However, if an agent can augment its vocabulary used to describe its knowledge base, it can use its terminology to communicate with other agents, and answer queries that it could not previously. In contrast to augmenting an agent ontology, the agent could retrieve the entire vocabulary required for communicate with another agent. This becomes inefficient when the agent interacts with the same agent more than once. In our context, our agents use an ontology to model their vocabulary and their knowledge base. Specifically, we take an ontology to be a formal structure that models concepts, relationships and entities. Then, the ability to evolve an agent's vocabulary enables it to reflect its environment, and ensure that its ontologies do not have to be remodelled in response to environment changes. However, such augmentation may incur costs, relating to the acquisition process and the search time required for inferring logical consequences from an agent's ontology. For this reason, our proposed technique attempts to reduce this by selecting the concepts and the neighbours to share with in an informed manner.

In more detail, this approach supports the automatic exchange of knowledge to augment agent based problem solving. It provides a method that reduces the complexity of an ontology compared with augmenting an agent's ontology with all knowledge in the environment, and selects domain-specific concepts that relate to an agent's ontology. While other techniques, such as those detailed below do augment an ontology, we focus on the analysis of an agent's ontology and compare it to a fragment that contains a shared representation of a concept. A query agent can therefore evaluate the relationship between each concept from the fragment and its ontology's domain. This approach enables agents to reduce the cost of mediated transactions by augmenting their ontology on demand.

We propose that our approach is appropriate for agents that provide services about a domain, and have a small ontology (approximately 50-200 concepts) that models an incomplete domain[2]. In our context, a service is the ability to provide and complete requests related to a specific domain. Agents in an environment that provide different services can benefit from incorporating new knowledge from other agents where their interest domains intersect. For example, suppose a set of emergency services have intersecting knowledge about the domain 'rescue'. When an emergency service agent (or query agent) cannot perform a rescue task alone, it can incorporate new domain knowledge from other specialist agents which support this task. In more detail, a specialist agent can send fragments to other agents about concepts in its ontology. For example, consider a query agent that requires a vehicle that can remove heavy rubble so that the hospital emergency service can rescue casualties from a collapsed building. In this case, the hospital emergency service agent's ontology is unlikely to contain knowledge about vehicles that can remove rubble. However, it is possible to learn this new information from a specialist agent so that it can organise the removal of the rubble.

The approaches presented by Bailin and Truszkowski [1], Afsharchi et al. [2], Wiesman and Roos [3], and Soh [4] enable their agents to augment their ontologies with new knowledge, when agents have different domain models representing the same domain. In contrast to our approach, these approaches require their agents to model the same domain. In particular, Bailin and Truszkowski's approach considers semantically equivalent representations, and Afsharchi et al. and Soh focus on the validation of the knowledge to be incorporated into the agent's ontology. These approaches augment an agent's ontology, however they augment their agent's ontology one concept at a time, which increases the overhead cost of retrieving the information. In addition to agent-based research, the Semantic Web community has produced work on evolving ontologies. In particular, the techniques presented by Flouris et al. [5] enable the evaluation of coherence and consistency. The work presented by Hasse and Stojanvoic [6] further explores the issue of consistency, by proposing techniques to resolve three types of inconsistency; structural, logical, and user defined inconsistencies. These techniques can be used with our approach in order to evaluate, locate and resolve inconsistent knowledge to be incorporated into an ontology.

Given this background, we have considered how to augment an agent's ontology with new knowledge, while analysing how to reduce the overhead cost involved with augmenting an ontology. Similarly to Afsharchi et al. and Soh, we also consider how to incorporate knowledge and select which knowledge has a higher priority, by considering which knowledge is contained in the majority of collab-

[1] School of Electronics and Computer Science, University of Southampton, United Kingdom, email:{hp07r, nmg, nrj}@ecs.soton.ac.uk

[2] An ontology that has an incomplete domain model does not model all concepts associated with its domain.

orating agents' ontologies. In contrast to the above approaches we augment our agent's ontology with a fragment of knowledge, as opposed to a single concept, in order to reduce the complexity of the knowledge that is required. This aims to satisfy our objective to reduce the overhead cost of regularly acquiring knowledge for repetitive queries. This approach is described in more detail in [7].

Our paper is structured as follows. In Section 2 we present an outline of how our query agent retrieves fragments, and in Section 3 we describe our technique to select concepts and relationships to augment a query agent's ontology. Then, in Section 4 we present our empirical evaluation of our technique. We conclude in Section 5.

2 Retrieving the Fragments

As previously discussed, our proposed approach enables an agent to evolve its ontology with new concepts related to its domain, so that it can collaborate and reduce the cost of inference associated with complex ontologies. In order to evolve a query agent's ontology, the agent is required to retrieve a fragment from a specialist agent's ontology modelling the required concept. To this end, Figure 1, shows three types of agent: 'query agent', 'specialist agent', and 'mediator'. We note three things about these agents. First, the query agent refers to its own ontology, which describes a specific domain. However, it models the domain only with partial knowledge due to its design and/or changing requirements. Second, the specialist agents also refer to their own ontologies, each of which specialises on a domain that intersects with the domain of the query agent. Third, the mediator can provide translation mappings, for the specialist agents, between a concept and a set of concepts.

In this scenario, the query agent is sent queries from a user. When the query agent receives a query that contains a concept that is not contained within its ontology, the query agent attempts to learn this new vocabulary from the specialist agents so that it can answer the query. In particular, figure 1 shows the interaction between the three agents where the query agent is required to learn a concept. In step 1 (and given our motivating example) a user sends our emergency rescue query agent a query to locate a vehicle that can move rubble from a collapsed building. This query contains a concept, in this example '`forkLiftTruck`', that the query agent ontology does not contain. In step 2, a request is broadcast to all specialist agents in the environment for knowledge about '`forkLiftTruck`'s. In step 3, the specialist agents request a translation between its concepts and the unknown concept, '`forkLiftTruck`'. This requested translation is sent in step 4. In this case only one specialist agent's ontology contains an equivalent concept to the unknown concept, '`forkLiftTruck`', and it sends this confirmation to the query agent in step 5. The query agent then requests the description in the form of a fragment based on the unknown concept, '`forkLiftTruck`', and the specialist agent responds with this fragment in steps 6 and 7. This fragment contains the concepts '`vehicle`', '`liftingCapacity`', '`building`', '`reachTruck`', '`handPalletTruck`', and '`truckMountedForklift`'. Once the query agent receives the definition of a '`forkLiftTruck`' it answers the user's query, shown in step 8. This enables the query agent to find a vehicle that can remove building rubble. Given this background, our approach focuses on the mechanism to select axioms to include into the query agent's ontology; additionally we also focused on the performance of the acquisition.

The query agent's approach is designed to select a set of concepts and relationships from a set of fragments that describes the required

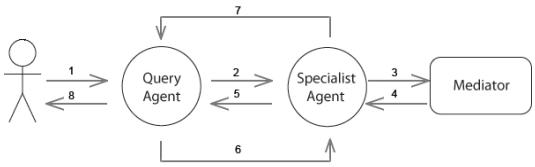


Figure 1. A sequence of messages between a user, query and specialist agent, and a mediator.

concept to augment into its ontology. This selection of concepts enables the agent to choose domain related knowledge, and limits the amount of knowledge augmented by the agent. The limitation of knowledge aims to reduce the potential reasoning complexity of an agent's ontology because the more complex an ontology is the more time is required to process inferred knowledge. It is important to note that this approach does not preclude the ability of the agent to complete complex tasks, as any required knowledge is never excluded. While evolving an ontology can benefit an agent as it can infer logical consequences from its ontology, a complex ontology can affect the time it takes to infer knowledge. This cost can be exponential depending on the structure and features contained within the ontology [8].

3 Collaborative Evolution of the Ontology

As discussed in the previous section, the query agent can obtain more than one fragment that possibly represents the required concept. Our main objectives are to reduce the overhead cost of regularly acquiring the same knowledge for repetitive queries, and reduce the complexity of using all the knowledge that is acquired. In order to do this we augment the query agent's ontology with a set of selected concepts and relationships representing the desired knowledge, and these components are stored as axioms in a query agent's ontology. In particular, we aim to maintain a similar granularity to the query agent's ontology while augmenting its ontology, to relate to the level of detail required for domain queries; this technique prevents an exponential growth of the depth of the agent's ontology, and increases the chance of retaining irrelevant concepts and relationships. Relating to the purpose of the agent, the granularity of an ontology is dependant on the agent's purpose, and agents that contain the same conceptualisations may have a different level of detail. For example, in our motivating scenario where a query agent wants to remove rubble, the agent requires information about only vehicles that can be used in rescue situations and will not need information about a '`reachTruck`' and '`handPalletTruck`' as these vehicles are for commercial environments. In order to achieve this we have a two stage process: merging the fragments; and selecting concepts and relationships from the merged fragment.

3.1 Merging the Fragments

The query agent can retrieve more than one fragment containing axioms relating to the requested concept. Our aim is to merge a subset of these fragments which are semantically similar to the agent's ontology, so that the query agent can select the subset of axioms from the merged fragments. This merging process removes fragments that do not relate to the agent's domain, redundant and conflicting axioms. This enables the query agent to incorporate knowledge that is contained within a greater number of agents, and therefore collaborate with a wider range of neighbours. The following process describes our chosen technique, and is used to merge the set of fragments:

1. Compare each fragment to the query agent's ontology, by requesting mappings from a mediator. If the fragment contains one other concept that matches the agent's ontology then it is deemed that the fragment and ontology have the same domain. In our example, the query agent's ontology concept 'capacity' maps to '`liftingCapacity`' which is contained in a retrieved fragment, and the concept '`vehicle`' is already contained in the query agent's ontology. Therefore the fragment detailed in Section 2 is deemed to relate to the query agent's domain. This step aims to remove fragments that do not represent the semantics of the requested concept. We assume that a fragment will contain concepts related to the ontology if they are domain related.
2. Generate the powerset of all axioms in the selected fragments. Discard sets which are inconsistent with the query agent's ontology, and select the largest set that is contained by the largest average number of agents.

This technique is used to select axioms that represent an agent's domain, and provide a set of axioms that do not conflict with the query agent's ontology.

3.2 Selecting Concept to the Merged Fragment

The above technique provides a merged fragment that represents the requested concept, so that the query agent can select a set of axioms from this fragment. From this, the selection component needs to select a set of axioms to use to augment the agent's ontology. To do this, we adopt a selection method that is similar to Seidenburg et al.'s [9] ontology segmentation technique, in that we both consider the role of hierarchical and relational classes. However, in contrast to Seidenburg et al.'s approach, which focuses on reducing the overall size of an ontology by selecting those axioms relating to one specific concept, our approach aims to reduce the number of axioms used to describe a specific concept. This retrieves a fragment that describes the context of a concept; this context is used for validating and not all of this context is required by the query agent. In order to maintain the query agent's knowledge, we model acquired axioms in a separate ontology, which is imported into the agent's instantiated ontology. This enables an agent to infer new and exploitable knowledge from its instances, and enable it to determine its instantiated concepts. For instance, in our example augmenting the query agent's ontology with a fragment about 'forkLiftTrucks' enables the agent to infer that a vehicle in its knowledge base can also lift the same amount as a forklift truck and therefore may be offered as a substitute in a critical situation. Our approach uses two steps to analyse these axioms: (i) **hierarchical**, and (ii) **relational** axiom selection techniques.

The **hierarchical selection** technique reduces the depth of a fragment, and provides the relational axiom selection technique with a set of concepts it can select from. Our hierarchical selection technique aims to reduce the depth of the fragment only if the depth of the query agent's ontology is smaller than the fragment's depth, otherwise it enables the relational selection algorithm to select from all of the fragment's levels. This technique calculates the mean average depth (from the root node to lowest child node) of the query agent's ontology and the fragment, and uses this to select the number of levels of classes to be selected from the fragment. Once the number of required levels has been selected, we calculate the number of times each concept in the fragment is referred to in the query agent's ontology. We then calculate the average number of times the concepts are referred to in each axiom, for each depth in the fragment. The selection process selects the number of required levels by selecting those with the highest average concept rating.

Relational selection aims to limit the number of relationships connected to the required concept. The number of properties and possible concepts have already been reduced by the *hierarchical selection* technique. The properties will be 'pruned' by the distance of properties in 'hops' away from the required concept, using a set threshold. This process ensures that the properties to be incorporated into the query agent's ontology are closely related to the required concept and its domain.

Once the hierarchical and relational selection processes have been performed, the selected axioms represent a shared set of axioms describing the required concept. These selected axioms are then added to the query agent's ontology.

4 Empirical Evaluation

In order to evaluate our approach, we performed an empirical investigation. We modelled a scenario where a query agent aims to answer a user's queries which relate to its ontology's domain about 'emergency rescue'. Our investigation compares four alternative approaches for the query agent, each of which processes the retrieved fragments (see Section 2). Our testing environment comprised of five query agents, one of which uses our approach (see Section 3), and four other query agents. These four query agents utilise the following four learning techniques:

1. *Learn-repeated* approach: learns all concepts and their relationships which are required more than once. This aims to offset the cost of learning concepts by considering how much the required concept is used.
2. *Learn-connected* approach: learns all axioms from fragments that are directly connected to the concept being queried. This is a comparable technique with the agent approaches discussed in Section 1.
3. *Learn-all* approach: learns all axioms in all of the fragments it locates. This technique is used to show that our approach's query agent has a lower ontology complexity than a query agent that learns everything.
4. *Learn-nothing* approach: learns nothing. This technique is used to show that our approach's query agent has a lower retrieval cost than a query agent that learns nothing.

Each of the query agents is instantiated with its own copy of the same ontology, and is given the same list of queries. This list of queries is generated by selecting a set of concepts related to a specific concept in the domain of 'emergency rescue' in sets of five, until there are one hundred queries. Also in the environment are ten specialist agents, five of which contain their own ontologies with the domain of 'emergency rescue', and the other five contain ontologies with domains unrelated to 'emergency rescue'.

The emergency rescue domain ontologies were derived from the AKTiveSA ontology[3] by the e-response project[4], which publishes ontologies for emergency rescue in the domains of Fire, Ambulance and Police. The non-emergency rescue ontologies are Beer[5], Brain

[3] AKTiveSA Ontology: `http://sa.aktivespace.org/aktivesa`
[4] e-Response Ontologies: `http://e-response.org/ontology/`
[5] Beer ontology: `http://www.purl.org/net/ontology/beer#`

Atlas[6], Bug reports[7], Charly Air Service[8] and Food[9]. After our agents have processed the one hundred queries, we calculate the costs involved for completing each query. These costs are measured in milliseconds, and include the time taken to: send messages across a network with a bandwidth of 2Mb, process an alignment, generate a fragment, and augment an agent's ontology. We also measure the complexity of the agent's ontology, this measure is calculated by the number of CPU clock ticks required to load and consistency check an ontology, using the Pellet reasoner[10]. In order to evaluate our approach we repeat our investigation fifty times to obtain a statistically significant average of each iteration. The results of our investigation are illustrated in Figures 2 and 3, which show the total cost (as described above) and ontology complexity, respectively. A confidence interval of 95% is indicated with error bars for each query.

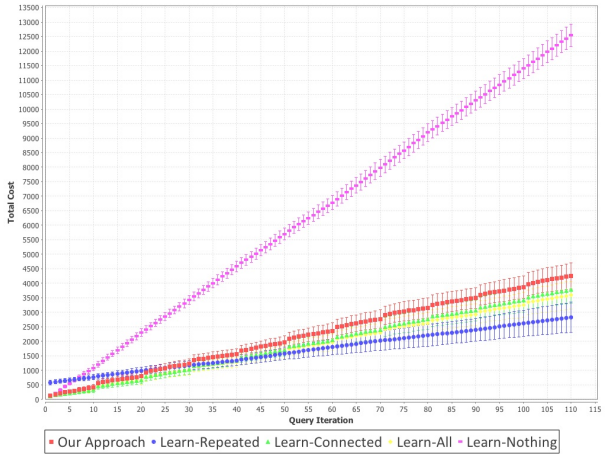


Figure 2. Cumulative total cost of alignment

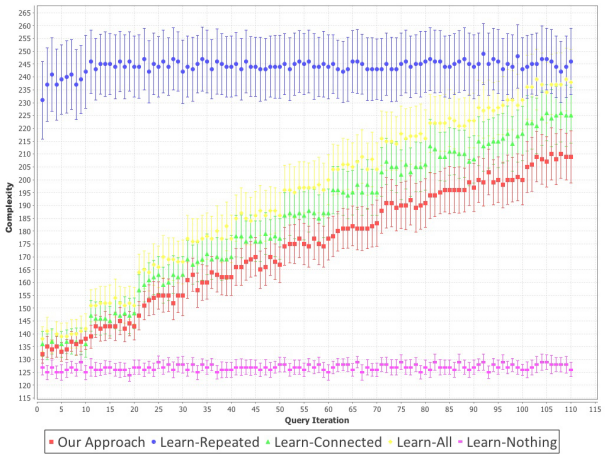


Figure 3. Ontology complexity

These figures identify four notable results. First, the results for the 'Learn-Nothing' approach demonstrate that while this technique keeps the ontology complexity consistently low, the total cost is highest of all techniques. Second, after all queries are completed, the ordering of the alternative query agent approaches is exactly reversed between the total cost and the ontology complexity. This illustrates that these two metrics are inversely linked, therefore we note that there is a trade off between the cost of selecting a fragment to augment a query agent's ontology and the complexity of a query agent's ontology. Third, that the overhead cost of learning specific repeated fragments, as used in the 'Learn-Repeated' approach, becomes profitable after approximately twenty five queries. This is because our approach does not know which queries will be repeated and the 'Learn-Repeated' knows of all repeated queries. Fourth, that the costs resulting from alternative approaches 2 and 3 are similar because their query agents communicate the same number of times.

When taken together, this shows that the 'Learn-Connected' and 'Learn-All' approaches have similar mean costs. More over, these alternative approaches have a lower mean total cost than our method. However, our approach's cost is within their confidence internal. Also, our approach has a lower mean complexity than these approaches because it selects concepts to augment into the query agent's ontology that relate to the its interest domain. This means our approach enables an agent to reduce the cost to acquire concepts that are regularly required, when compared to the 'Learn-Nothing' approach. It also reduces the complexity of our query agent's ontology by augmenting it with selected concepts and relationships, compared with the 'Learn-Repeated', 'Learn-Connected' and 'Learn-All' approaches.

5 Conclusions

We have presented a technique that enables an agent to automatically retrieve knowledge and augment its ontology in order to reduce the cost of acquiring regularly required concepts. Our technique enables a query agent to select concepts and relationships from a fragment to augment into its ontology. We hypothesised that with this selection process we could reduce the complexity of augmenting a query agent's ontology, compared with a number of standard alternatives. Our investigation shows that a query agent can indeed reduce the cost to acquire concepts that are regularly required, compared with learning nothing. It also reduces the complexity of the query agent's ontology by augmenting it with selected concepts and relationships which are related to its domain. We also hypothesise that to decrease the agent's ontology's complexity further we can discard irrelevant information, and this will be the focus of our future work. Specifically, this focus aims to identify the value of the knowledge contained in an agent's ontology so that it can select which concepts and relationships to 'forget'.

[6] Brain Atlas Domain ontology: `http://twiki.ipaw.info/bin/view/Challenge/OntoGrid`

[7] Bug Reports ontology: `http://www.cs.cmu.edu/ãnupriya/bugs`

[8] Charly Air Service ontology: `http://www.fo-ss.ch/simon/DiplomaThesis/Ontologies/CharlyAirService.owl#`

[9] Food ontology: `http://www.hut.fi/ťomik/food#`

[10] Pellet: `http://clarkparsia.com/pellet/`

REFERENCES

[1] Bailin, S., Truszkowski, W.: Ontology negotiation between intelligent information agents. The Knowledge Engineering Review **17**(01) (2002) 7–19

[2] Afsharchi, M., Far, B., Denzinger, J.: Ontology-guided learning to improve communication between groups of agents. Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, Hakodate, Japan (2006) 923–930

[3] Wiesman, F., Roos, N.: Domain Independent Learning of Ontology Mappings. Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems, New York City, New York, USA (2004) 846–853

[4] Soh, L.: Multiagent, Distributed Ontology Learning. Working Notes of the 2nd AAMAS OAS Workshop, Bologna, Italy (2002)

[5] Flouris, G., Huang, Z., Pan, J., Plexousakis, D., Wache, H.: Inconsistencies, Negations and Changes in Ontologies. In: Proceedings of the National Conference on Artificial Intelligence. Volume 21 (2006) 1295

[6] Haase, P., Stojanovic, L.: Consistent Evolution of OWL Ontologies. Proceedings of the 2nd European Semantic Web Conference, Heraklion, Greece (2005)

[7] Packer, H., Gibbins, N., Payne, T., Jennings, N.: Evolving Ontological Knowledge Bases through Agent Collaboration. European Workshop on Mulit-Agent Systems, Bath, UK (2008)

[8] Reynolds, D. and Thompson, C. and Mukerji, J. and Coleman, D.: An assessment of RDF/OWL modelling. Technical Report HPL-2005-189, October 28, 2005

[9] Seidenberg, J., Rector, A.: Web ontology segmentation: analysis, classification and use. In: Proceedings of the 15th International Conference on World Wide Web, New York, NY, USA, ACM (2006) 13–22

Ontology Evolution in Legal Reasoning: A study of ontology interpretation (Extended Abstract)

Andrew Priddle-Higson

Abstract.
We are researching the problem of open texture in legal reasoning and how it leads to the evolution of legal ontologies. Concepts in law are open-textured, they can't be uniquely matched to common sense concepts which describe real-world events. The argumentation that occurs in many legal cases is about the meaning of legal concepts. These legal concepts evolve through this argumentation. Our aim is to build computational models of this legal reasoning. We argue for the importance of ontologies for meta-knowledge in modelling this argumentation. We also argue that this work has relevance for ontology matching.

1 Legal Reasoning and Open Texture

Legal reasoning is not simply a matter of identifying the subsumptions between legal concepts and real-world concepts. Legal concepts are not precisely defined enough for this: the relationship between them and the real-world is not exact. H.L.A. Hart, who was the first to explicitly describe this phenomena[1] [5], gives the example of a legal rule governing a park: "No Vehicles in the Park". The rule seems clear enough, but what about the situation where a child wants to ride his bike in the park? Does the definition of "vehicle" in the rule include bikes? Hart argued that legal rules have a core meaning which we agree on, e.g. a car or a lorry is a vehicle, but there is also a fringe of uncertain cases where we are not sure whether we can, or should, apply the rule. These cases are hard legal cases, where lawyers must argue for their preferred interpretation and judges must best resolve the open-texture.

There is an analogy between the problem faced by a lawyer using the law and that faced by an agent using an ontology. The lawyer did not create the law, but must find a match between the legal concepts and the real-world situation. For instance, a lawyer must argue why "vehicle" should or should not include "bike". Similiarly, an agent must use an ontology (which they probably did not create) and find matches between it and the ontologies used by other agents. For instance, an agent might have to find a match between "automobile" and "motor vehicle". The legal problem can be viewed as a matching problem between an abstract legal ontology and a commonsense real-world ontology.

Legal reasoning is different from the reasoning that currently occurs in ontology matching because of its reflective nature. Lawyers do not just argue about the meaning of legal concepts, they can also argue about how legal concepts acquire meaning. For instance, in the case of *James Buchanan and Co Ltd v. Babco Forwarding and Shipping (UK) Ltd*[2] lawyers argued over the meaning of "other charges incurred in respect of the carriage of the goods". In trying to determine a meaning for the phrase, they had to argue about how the phrase could acquire meaning in the context of the case. The law was based on an international treaty, and there was a French language version of the law, in addition to the version enacted in English law. The lawyers argued about whether the meaning of the French law could be used to help interpret the meaning of the English law. The methods they were using to determine the meaning of the term were part of the domain of argumentation. Contextual information is already used to aid ontology matching, e.g. the S-Match system [3], what differs here is the reasoning *about* the rules for using those contextual sources.

[1] Although Hart's work is influenced by philosophical work on meaning in natural language by Wittgenstein and Waissmann.

2 Ontology Evolution

Legal rules and concepts evolve because of the argumentation about their meaning. This is particularly obvious in a common-law legal system in which precedents are binding. So if a case decided that "vehicle" should not include a child's bike, in future children would be safe to cycle through the park. The meaning of "vehicle" has changed in this context to exclude bikes, at least when they are ridden by children.

The arguments about the meaning of a legal concept depend upon meta-theories of knowledge and meaning. The arguments involve meta-ontological concepts such as "includes" or "depends upon". For instance, a lawyer might argue: "the definition of vehicle depends upon the maximum speed the object is capable of, since vehicles travelling slower than 5 miles per hour aren't a danger to the public." This argument contains an assertion about what the definition of vehicle should be. This can be viewed as an assertion about the definition of the vehicle concept:

$Vehicle(x) \rightarrow maxspeed(x) > 5mph$

This definition is justified by the argument that the concept of "vehicle" was intended to prevent dangerous situations and that a vehicle whose speed is less than 5 miles per hour is not dangerous. But another lawyer could challenge this justification. They could argue that the law was intended to prevent bikes from damaging the grass, so any bike should be banned from the park.

We can see from this example that evolution of the meaning of the legal concept is driven by the arguments that are created by the lawyers and that are accepted by the judge. The lawyers are motivated to create these arguments because they want to win the case, the judge is motivated to find the "best" legal solution to the problem. Naturally the judge is the hardest agent to model in this case, since we have no algorithm to determining the "best" legal solution. However, the lawyers can be modelled as agents which have the goal

[2] [1978] A.C. 141

of winning the case for their client, they must find "good" legal arguments from the various contextual sources which they have available. The challenge of developing a computational model is to give some operational definition of what good is and to prepare the background knowledge necessary to generate the arguments.

3 Computational Model

We are using work on contextual logics [1] as a basis for our formalisation of legal reasoning. Our computational model is based upon the idea of legal reasoning as contextual theory construction. An agent has to create a $Theory$ context which contains arguments which justify an interpretation of the open-textured legal rule. The arguments are based upon the content of other contexts and the bridge rules for composing them.

For example, an argument about the meaning of vehicle might look like:

$$\begin{aligned} Park &: \forall x.Vehicle(x) \wedge inPark(x) \rightarrow Fine(rider(x)) \\ RealWorld &: bike(bike01) \wedge inPark(bike01) \\ Commonsense &: \forall x.bike(x) \rightarrow Vehicle(x) \\ &\vdash Fine(rider(bike01)) \end{aligned}$$

This argument states that since: the $Park$[3] context states that if a vehicle is in the park then its owner should be fined; the $RealWorld$ context states that the bike $bike01$ is in the park; and a $Commonsense$[4] context states that any bike is a vehicle; then we can conclude that the the bike rider should be fined.

The problem of developing a theory is one of taking various contextual background knowledge sources and using them to create a set of arguments which justifies either applying the law or not in a particular real-world situation. We would like to develop a system which will take in relevant background knowledge to a legal case, and a description of the real-world situation and the law; the system will then try to find arguments for both sides from the background knowledge that would justify applying the law to the real-world situation. The process of finding arguments uses the contextual and meta-level reasoning to argue about how the law can (and should) be interpreted.

The main challenge is to develop the ontologies to describe the meaning of legal rules and the meta-level reasoning required to reason about meaning. The ontologies about meaning must also relate with the facts of a legal case, so we can bridge between what happened and an argument about how the law should be interpreted.

4 Related Work

Our work is closely related to work on contextual reasoning within A.I. and work on open-texture within A.I. and Law. We have been using a model of context developed by researchers at the University of Trento [2]. However the examples normally used to motivate research into contextual logics, e.g. the magic box example [2], are not very complex; there is a large gap between them and real-world contextual reasoning, which our research is helping to bridge.

There has been some research into the problem of open-texture within the AI and Law community. Most of this work has focused upon the use of defeasible reasoning[5] to represent and reason about the exceptions to a legal rule. The most similiar work was done by Andreas Hamfelt [4], who used meta-logic programming to formalise the multiple levels of legal rules and how these can determine the interpretation of a legal rule. In particular, he used a meta-ontological predicate "Meaning" to formalise the relationship between a natural-language legal-rule and a formal representation of the rule.

[3] a context which represents the content of a park bylaw

[4] representing a context of background commonsense knowledge which a lawyer can appeal to

[5] related to the fields of non-monotonic logic and argumentation

5 Conclusion

We have presented a brief description of our research into modelling ontology evolution in law and how it relates to problems in ontology matching. Our model is based upon contextual reasoning in which a legal agent must reason about how bridge rules between contexts can be used to create a theory regarding the interpretation of an open-textured legal rule.

REFERENCES

[1] Bouquet P. Benerecetti M. and Ghidini C. Contextual reasoning distilled. *Journal of Experimental Artificial Intelligence*, 12(3):279–305, 2000.

[2] C. Ghidini and F. Giunchiglia. Local Models Semantics, or contextual reasoning= locality+ compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.

[3] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-Match: an Algorithm and an Implementation of Semantic Matching. *Lecture Notes in Computer Science Volume 3053*, pages 61–75, 2004.

[4] A. Hamfelt. Formalizing multiple interpretation of legal knowledge. *Artificial Intelligence and Law*, 3(4):221–265, 1995.

[5] HLA Hart. The Concept of Law, Clarendon Law Series, 1961.

Detecting unknown word senses using concept dictionary

Yoshimi Suzuki [1] and **Fumiyo Fukumoto**

Abstract. In this paper, we present a method for detecting unknown word senses using a concept dictionary and newspaper articles. It is very important to detecting unknown word senses for document classification, information retrieval, information extraction, etc. Although for extracting similar word pairs the methods which use similarity of case structure between two words are used, comparison between similarity of two words suffers word sparseness problem. Especially, it is necessary to solve this problem for detecting word senses of proper nouns which are not listed in the dictionary. The proposed method used hierarchical semantic features of a concept dictionary in order to deal with this problem. We performed some experiments in order to confirm effectiveness of the method.

1 Introduction

In newspaper, web pages, patent documents, etc., many different proper nouns frequently appear, and furthermore, new proper nouns are generated in these documents day after day. Most of proper nouns did not given in dictionaries, therefore we have to detect their senses for some applications of natural language processing: thesaurus construction and the rest. For recognizing proper nouns, named entity extraction and named entity recognition are studied frequently. For named entity extraction and recognition, machine learning techniques are used in order to categorize each named entity into some types [1]: ORGANIZATION, PERSON, LOCATION ARTIFACT, DATE, TIME, MONEY and PERCENT. Sekine proposed extended 200 named entity categories [2]. The extended categories are effective for QA application. But the types may be coarse for detecting unknown word senses.

Thesaurus construction is also studied actively. Many studies use context information for similar noun pairs [3], and many similar noun pairs of common nouns can be extracted accurately. But it is difficult to extract pairs of proper noun and common noun which are semantically similar. Because some proper nouns do not frequently appear and we can extract very few information of dependency relationship for the proper nouns.

In this paper, we present a method for detecting unknown word senses using concept dictionary and newspaper articles. Although the methods which use similarity of case structure between two words are used for extracting similar word pairs, calculating similarity between two words suffers word sparseness problem. Especially, it is necessary to solve this problem for detecting word senses of proper nouns which are not listed in the dictionary.

The proposed method used hierarchical semantic features of a concept dictionary in order to deal with this problem. We performed some experiments in order to confirm effectiveness of the method.

[1] University of Yamanashi, Kofu 400-8511, Japan, email: ysuzuki@yamanashi.ac.jp

2 System Overview

Figure 1 illustrates the overview of our system. Our system consists of 5 steps which are described in the following list.

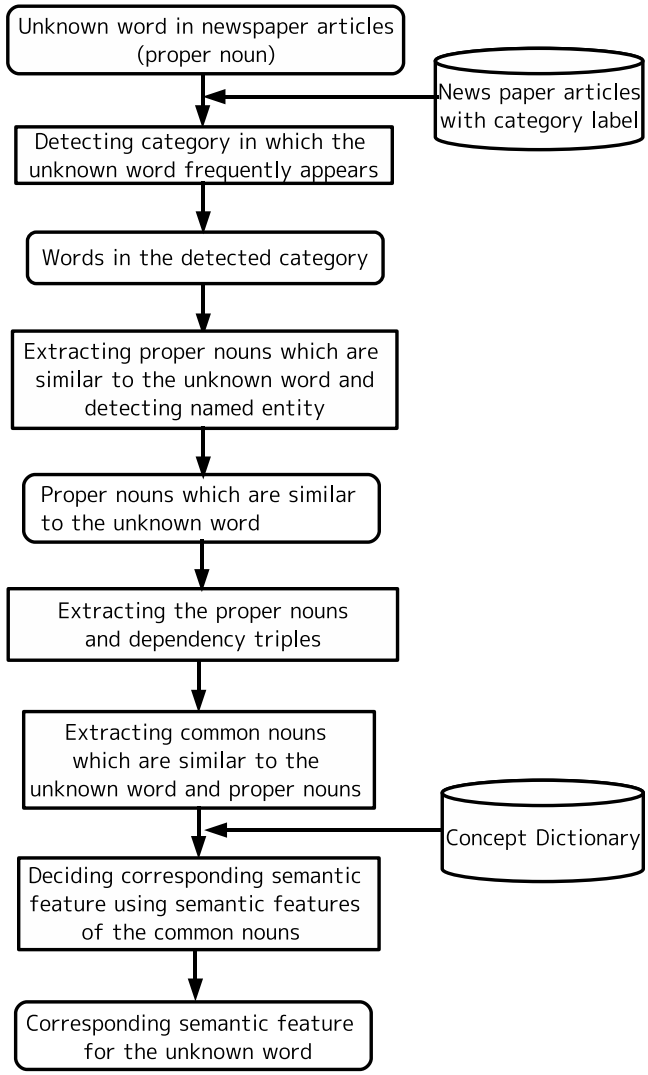


Figure 1. System Overview

1. Detecting the category in which the target unknown words appear, and extracting dependency pattern which include target unknown word.

2. Extracting similar proper nouns of the target word from the detected category. Then extracting named entity information of the target unknown word using a CaboCha [4], and extracting words whose dependency pattern of the words which are categorized into same named entity category.
3. Extracting dependency triples of the extracted proper nouns.
4. Extracting common nouns which are similar to the unknown word and its similar proper nouns.
5. Deciding corresponding word senses using semantic features of the extracted common nouns.

We explain our system step by step with an example: "Tigers". "Tigers" is not listed the dictionary, but must be "baseball team".

3 Detecting Category

The aim of our study is to detect the appropriate semantic feature of target unknown words. Some proper nouns have different meanings in different categories. For example, "Tigers" illustrates names of baseball teams and also a name of musical band. Therefore "Tigers" has different dependency pattern in different category.

Table 1 shows frequency of appearance of "Tigers" and similar proper nouns in each category of the newspaper articles. In Table 1, most of "Tigers" appear in Sports category. Therefore we used newspaper articles in Sports category for deciding semantic feature of "Tigers" appeared in Sports.

Table 1. Frequency of appearance in each category

Category	Tigers	Giants	Twins	Mariners
1st page	70	65	20	304
2nd page	26	13	0	25
3rd page	28	2	7	30
Sports	1207	2163	1269	4942
Home	5	2	2	3
Science	0	0	1	1
Commentary	26	9	0	38
Economy	36	3	0	17
Show biz	20	2	0	7
Int'l	35	31	34	193
Local	406	381	70	760
Editorial	3	9	0	10
General	96	100	12	151
Special	45	123	28	191
Reading	6	7	0	4
Culture	0	0	0	2

It is necessary to extract word candidates with category information, because semantic features of concept dictionary are classified without their categories.

4 Extracting proper nouns which are similar to the unknown word

Some proper nouns do not frequently appear in even if large corpora. Table 2 shows frequencies of the team names of Japan Professional Baseball in the newspaper articles of The Mainichi Newspapers (2000). Although the 6 teams in Table 2 are members of the same league, "Tigers"appears frequently, however "Swallows" appears 3 times. Therefore in order to deal with the proper nouns like a "Swallows", we have to extract proper nouns which are same members of cordinate terms.

Table 3 illustrates the extracted proper nouns which are similar to "Tigers". In table 3 All of top 5 are baseball teams.

Table 2. Frequencies of 6 team names of Japan Professional Baseball in the articles of The Mainichi Newspaper (2000)

Team Names	frequency
Giants	182
Tigers	398
Dragons	22
Carp	34
Swallows	3
Baystars	49

Table 3. Extracted proper nouns for "Tigers" (top 5)

rank	proper noun	similarity
1	Orix (baseball team)	0.448
2	Seibu (baseball team)	0.439
3	Yokohama (basebal team)	0.417
4	Chunichi (baseball team)	0.416
5	Kyojin (baseball team)	0.400

4.1 Dependency relationship

Dependency information is used for extracting semantic similar pairs. For example, Lin proposed "dependency triple" [5]. A dependency triple consists of two words: w, w' and the grammatical relationship between them:r in the input sentence. $||w, r, w'||$ denotes the frequency count of the dependency triple (w, r, w'). $||w, r, *||$ denotes the total occurrences of $w - r$ relationships in the corpus, where $*$ indicates wild card.

When it applies for the unknown word "Tigers", (quit, object, Tigers) is obtained.

But most of the unknown words do not appear frequently, then we have to use hierarchical semantic feature for smoothing technique.

5 Extracting dependency triples of the proper nouns

In order to extract corresponding common nouns, we extract dependency triples of the extracted proper nouns. Table 4 illustrates examples of dependency triples of the extracted proper nouns. Using some extracted proper nouns, many types of dependency triples are extracted.

Table 4. Examples of dependency triples of the extracted proper nouns

w	r	w'
Chunichi	subject (*ga*)	escape (the cellar)
Hanshin	subject (*ga*)	rise (to second place)
Seibu	subject (*ga*)	run away
Taiyo	object (*wo*)	steamroller
Orix	to (*he*)	transfer

6 Extracting common nouns

We extract common nouns which are similar to the unknown word and the extracted proper nouns. Table 5 shows examples of extracted common nouns for "Tigers".

7 Detecting corresponding semantic features

Finally, candidates of corresponding semantic features of the unknown word using the concept dictionary.

Table 5. Extracted common nouns for "Tigers" (top 5)

rank	common noun	similarity
1	team	0.382
2	delegate	0.321
3	player	0.313
4	excellent team	0.185
5	horse	0.157

7.1 Hierarchical semantic features

We used hierarchical semantic features made by EDR [6]. Table 6 shows hierarchical semantic features of "baseball team".

Table 6. Hierarchal semantic structure (baseball team)

depth level	code	label
6	3c1e0f	team
5	444999	colleague, team mate
4	30f6b1	associate
3	3cfacc	group
2	3aa912	active object
1	3aa911	human being
0	3aa966	concept

8 Experiments

We confirmed that an unknown word "Tigers" correspond to baseball team by using our method.

8.1 Experimental setup

We used newspaper articles of The Mainichi Newspapers (from 1991 to 2004, written in Japanese). There are about 500 thousands articles. Unknown words are selected from newspaper articles of The Mainichi Newspapers (from 2000 to 2004, written in Japanese). We used concept dictionary by EDR. It has about 410 thousands concept classification records.

8.2 Experimental results and Discussion

Although we use very few test data, we can detect an unknown word sense using our method. At each step, we evaluate the results. All test data frequently appear in one category, by contrast, in other categories they do not frequently appear. When the unknown word appeared frequently in some categories, the unknown word has some different meanings. For example, "Tigers" meant it any place other than a baseball team. Therefore, we found that it was important to use the articles which are classified into categories.

9 Conclusion

This paper proposed a method to detect unknown word senses using concept dictionary. We used concept dictionary by EDR, categorized newspaper articles, and we showed the possibility of detecting unknown word senses using the proposed method. In the future, we plan to the following.

- Using test data to confirm the method is effective.
- Using machine learning technique.
- Using Web documents for collecting .
- Detecting unknown word senses in patent document

REFERENCES

[1] Satoshi, Sekine., Hitoshi, Isahara., "IREX: IR and IE Evaluation project in Japanese", In Proceedings of the Forth International conference on Language Resources and Evaluation, 2000.

[2] Satoshi, Sekine., Chikashi Nobata., "Definition, dictionaries and tagger for Extended Namd Entity Hierarchy", In Proceedings of the Forth International conference on Language Resources and Evaluation, pp.1977-1980, 2004.

[3] Hagiwara, M., Ogawa, Y., Toyama, K.: Selection of Effective Contextual Information for Automatic Synonym Acquisition. In Proc. of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, 353-360 (2006)

[4] Kudo, T. and Matsumoto, Y., "Japanese Dependency Analysis using Cascaded Chunking", CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002, pp.63-69, 2002.

[5] Dekang, Lin., "Automatic Retrieval and Clustering of Similar Words", Proceedings of 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics Proceedings of the Conference, pp.768-774, 1998.

[6] EDR ELECTORONIC DICTIONARY VERSION 2.0 TECHNICAL GUIDE, National Institute of Information and Communications Technology (1996)

Dealing with Uncertainty Issues in Complex Ontology Matching

Ying Wang[1] and **Weiru Liu**[1] and **David Bell** [1]

Abstract. Ontology mapping is one of the most important tasks for ontology interoperability and its main aim is to find semantic relationships between entities of two ontologies. However, most of the current techniques suffer from some kind of drawbacks as listed below: (a) most of them only consider 1:1 mappings; (b) most of them do not consider the importance of uncertainty in ontology mapping. In this paper we consider the following two issues that have been the focus of our ongoing research: (a) how to produce complex mappings (m:1 or 1:m and m:n) and (b) how to deal with uncertainties in the process of ontology mapping.

1 INTRODUCTION TO THE PROBLEM

Research and development on ontology mapping (or matching) has attracted huge interests and many mapping methods have been proposed. Comprehensive surveys on recent developments of ontology mapping can be found in [10, 11].

Considerable efforts have been devoted to implement ontology mapping systems, especially 1:1 mappings. However, complex mappings are also pervasive and important in real world applications. In [10], an example was given to illustrate the importance of complex mappings in schema mapping research. We think that the same issue exists in ontology mapping and the example is applicable to ontology mapping. Let us take a look of the example. Given two ontologies O_A and O_B, they contain different entities respectively: **Book** and **Publisher** in O_A; **Title** and **Name** in O_B. It is clear that entities {**Book, Publisher**} of O_A should be matched to {**Title, Name**} of O_B.

Another aspect is that most of the earlier works in this area did not consider uncertainty or imprecision occurred during a mapping, however, in most cases, the mappings between entities produced are imprecise and uncertain. For instance, most automatic ontology mapping tools use heuristics or machine-learning techniques, which are imprecise by their very nature. Even experts are sometimes unsure about the exact matches between concepts and typically assign some certainty ratings to a match [2]. So a matching result is often associated with a weight which can express how close the two entities are as a match. The needs to consider uncertainty in a mapping began to emerge in a number of papers (e.g., [8, 1, 9, 4, 13]) in which Dempster Shafer theory, Bayesian Networks, and rough sets theory are used to deal with different aspects of mapping or ontology descriptions (e.g., concept subsumptions).

The rest of the paper is organized as follows. Section 2 presents the set-inclusion based approach we proposed for dealing with complex matching. Section 3 describes the clustering-based approach we developed for handling uncertainties in ontology mapping. Section 4 concludes the paper.

[1] School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, BT7 1NN, UK , email: {ywang14, w.liu, da.bell}@qub.ac.uk

2 A SET INCLUSION BASED ONTOLOGY MAPPING APPROACH

Before we introduce this new ontology mapping approach, we first describe a new method to represent entities in ontologies. Traditionally, the concept names of entities are used directly in mapping. This representation method does not consider the hidden relationships between concept names of entities, so it cannot reflect the complete meaning of the concept names of entities. Here we explore a new representation method for entities. For the multi-hierarchical structure of ontology, we observe that for each concept in this concept hierarchy, its complete meaning is described by a set of concept names. In other words, there is a kind of *inclusion relationship* among these concepts. So for any concept name of entity C in an ontology, we can represent it by a new method as follows. First, we find the branch which has the concept C. Second, we collect those concepts along the path between C and the root node to form a set. We use this new set to represent C.

Once each entity is represented by a set of words, we compute the similarities between entities. Here, we choose the *Linguistic-based matcher* (which uses domain specific thesauri to match words) and the *Structure-based matcher* (which uses concept-hierarchy theory) to compute similarities (we utilize Linguistic-based matcher because the performance of this matcher is good for similar or dissimilar words. Please refer to [12] for details).

As a result, we obtain a set S_1 consisting of mapping candidates such that from each entity in ontology O_1, a similarity value is obtained for every entity in ontology O_2. Following this, we select the best mapping entity in O_2 for each entity in O_1 and these best mapping results constitute another set S_2. In S_2, we search all the mapping results to see if there exist multiple source entities in O_1 that are mapped to the same target entity in O_2. If so, we apply a new algorithm based on Apriori algorithm [3] to decide how many source entities in O_1 should be combined together to map onto the same entity in O_2.

We use the OAEI 2007 Benchmark Tests and we now compare the outputs from our system (denoted as SIM) to the results obtained from *ASMOV, DSSim, TaxoMap* and *OntoDNA* algorithms which were used in the 2007 Ontology Alignment Contest [2] in which almost all the benchmark tests describe Bibliographic references and the details are given in Table 1. In Table 1, p for precision, r for recall, f for f-measure. Our study shows that this method significantly improves the matching results as illustrated in our experiments.

[2] http://oaei.ontologymatching.org/2007/results/

Table 1. Comparison of Experiment Results

Datasets	SIM			ASMOV			DSSim			TaxoMap			OntoDNA		
	p	r	f	p	r	f	p	r	f	p	r	f	p	r	f
101	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
103	100	100	100	100	100	100	100	100	100	100	34	51	94	100	97
104	100	100	100	100	100	100	100	100	100	100	34	51	94	100	97
203	100	100	100	100	100	100	100	100	100	NaN	0.00	NaN	94	100	97
204	86	84	85	100	100	100	96	91	93	92	24	38	93	84	88
205	47	44	46	100	100	100	94	33	49	77	10	18	57	12	20
208	86	83	85	100	100	100	95	90	92	NaN	0	NaN	93	84	88
209	49	41	45	92	90	91	91	32	47	NaN	0	NaN	57	12	20
221	82	82	82	100	100	100	100	100	100	100	34	51	93	76	83
222	89	92	91	100	100	100	100	100	100	100	31	47	94	100	97
224	100	100	100	100	100	100	100	100	100	100	34	51	94	100	97
225	100	100	100	100	100	100	100	100	100	100	34	51	94	100	97
228	100	100	100	100	100	100	100	100	100	100	100	100	53	27	36
230	73	90	81	99	100	99	97	100	98	100	35	52	91	100	95
231	100	100	100	100	100	100	100	100	100	100	34	51	94	100	97
232	82	82	82	100	100	100	100	100	100	100	34	51	93	76	84
233	52	52	52	100	100	100	100	100	100	100	100	100	53	27	32
236	100	100	100	100	100	100	100	100	100	100	100	100	53	27	32
237	93	97	95	100	100	100	100	100	100	100	31	47	94	100	97
239	88	100	94	97	100	98	97	100	98	100	100	100	50	31	38
241	58	58	58	100	100	100	100	100	100	100	100	100	53	27	32
246	88	100	94	97	100	98	97	100	98	100	100	100	50	31	38
301	43	45	44	93	82	87	82	30	44	100	21	35	88	69	77
302	34	53	42	68	58	63	85	60	70	100	21	35	90	40	55
304	51	49	50	95	96	95	96	92	94	93	34	50	92	88	90

Overall, we believe that the experimental results of our system are good. Although on individual pair of ontologies, our results are less ideal than the *ASMOV* system and *DSSim*, however, our results are better than *TaxoMap* system and *OntoDNA* system on most pairs of matching. The performances of these three different approaches, i.e., *ASMOV, DSSim* and our system *SIM* are good for almost the whole data set from Test 101 to Test 246, but our system does not perform well for Test 205, Test 209, Test 233 and Test 241. The performance of all these five systems are not very good for the data set from Test 301 to Test 304. Below we analyze the reasons for this.

For Test 101 vs 103 and vs 104, the two ontologies to be matched contain classes and properties with exactly the same names and structures, so every system that deploys the computation of similarities of names of entities can get good results. Test 201-210 describe the same kind of information as other ontologies, i.e. publications, however, the class names in them are very different from those in the reference ontology Test 101, especially Test 205 and 209, so our system does not obtain good results. The structure of Test 221-247 have been changed although the linguistic features have been maintained, the performance of our system has been affected. Our method is based on the hierarchical structure of an ontology, but for Test 233 and Test 241, these two ontologies have only one layer. When computing the similarity between two concepts in Test 233 and Test 101, such as **MastersThesis** in Test 233 and **MastersThesis** in Test 101. First, our method extends **MastersThesis**. Test 233 only has one layer, so **MastersThesis** can not be changed. Test 101 has three layers, so **MastersThesis** is extended to {**MastersThesis, Academic, Reference**}. The similarity value is reduced and does not reflect the true similarity between these two concepts.

Test 301-304 are real-life BibTeX ontologies which also include different words compared to Test 101 describing publications so the results are similar to Test 205, so we do not get good similarity results from this data set. However we still find some complex mappings (m:1) by using our algorithm to discover the best mapping results, such as for Test 302 vs Test 101, we get {**Collection, Monograph, Book**} mapping to **Book**.

3 CLUSTERING-BASED APPROACH TO COMBINING UNCERTAIN OUTPUTS FROM MULTIPLE ONTOLOGY MATCHERS

We propose a clustering-based approach to combining outputs from multiple ontology matchers (**CCM**). We consider complex mappings between two ontologies O_1 and O_2 which are encoded in OWL. First, we partition entities in ontology O_1 based on *average-linkage clustering algorithm*. This algorithm uses similarity values between entities to do the partitioning, the similarity values are obtained by integrating Lin's matcher [7] (which uses domain specific thesauri to match words) and structure-based method to compute the similarities between entities of O_1 (we utilize Lin's matcher because the performance of this matcher is good for similar or dissimilar words. Please refer to [12] for details). As a result, the similar entities in O_1 are clustered together. Second, for each entity e_{2j} in ontology O_2, we try to find the most appropriate cluster C_{1i} in the collection of clusters created from ontology O_1. Cluster C_{1i} is regarded as the most appropriate for e_{2j} if the similarity value between e_{2j} and C_{1i} is the largest. Third, we deploy four different matchers to calculate the similarity values between a cluster from O_1 and an entity in O_2. We choose several matchers because one matcher analyzes only some aspects of the hypothetical relation between two terms and may lack or omit important information about the relationship between entities [1]. Therefore, if we use more than one matcher, these matchers can complement each other and capture more features about the relationship between entities. Finally, since each match gives a mapping that is not absolutely certain, we apply Dempster-Shafer theory to combine the matching outputs from these four matchers.

We choose two pairs of ontologies, one is **Test 101-205** and another is **russia12**[3] that describe tourism information of Russia. These ontologies are well-known for ontology alignment tests. Their sizes are moderate. To evaluate the mapping quality, here we employ the metric of *correctness* and *f-measure*.

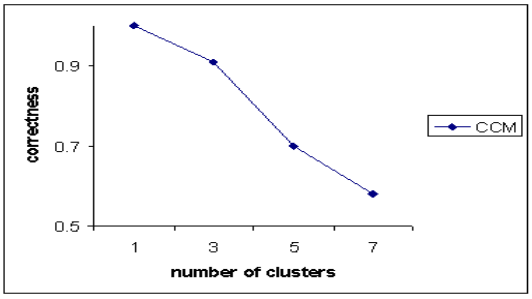


Figure 1. Test 101-205

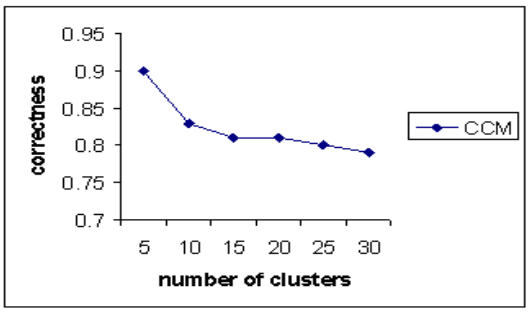


Figure 2. russia12

Both of Figure 1 and Figure 2 show the variation of correctness along with the number of the clusters in cluster mappings. From these two figures, we can see that when the number of clusters increases, the correctness of the cluster mapping decreases. For Figure 1, the names and structures of entities in these two ontologies are very different, so the downward trend of curved line is quick. For Figure 2, the names and structures of entities in these two ontologies are very similar, so the downward trend of CCM is slow when the number of clusters increases.

Table 2. Comparison of Experiment Results

Datasets	approach	number	correctness	f-measure
russia12	CCM	13	0.82	0.30
russia12	BMO	13	0.84	0.56
russia12	PBM	13	0.57	0.65

In Table 2, the comparison results of ontology mapping quality

[3] http://www.aifb.uni-karlsruhe.de/WBS/meh/foam/

and the partitioning quality of CCM, BMO [5] and PBM [6] are presented. The number of cluster mappings is 13. Overall, we believe the DS combination rule is effective although the *f-measure* of CCM is not very good. One of the reasons is that although we utilize a combination method which combine Lin-based matcher and structure method together to compute the similarity between entities in O_1, the results of similarity are still not very good. Another reason is that the matchers we used are based on linguistic features of entities of ontology and they can only handle the problem of mapping from one aspect, meanwhile the similarity results obtained from internal matchers are not very accurate, so when we combine these results by the DS combination rule, some useful but different results are left out.

4 CONCLUSION

Ontology mapping is a difficult task. So for our future work, on the one hand, we will continue improving our current proposed approaches, especially for complex mapping and how to deal with inconsistency produced by different matchers. On the other hand, we will continue investigating the uncertainty issue in ontology mapping and consider how to use different uncertainty theories to deal with different situations in ontology mapping.

REFERENCES

[1] P. Besana., 'A framework for combining ontology and schema matchers with dempster shafer', in *the International Workshop on Ontology Matching (OM'06), collocated with the 5th International Semantic Web Conference (ISWC'06)*, pp. 196–200, (2006).

[2] N. Choi, I.Y. Song, and H. Han, 'A survey on ontology mapping', *In SIGMOD Record*, **35**, 34–41, (2006).

[3] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2000.

[4] M. Holi and E. Hyvoen, 'Modeling degrees of conceptual overlap in semantic web ontologies', in *the International Workshop on Uncertainty Reasoning for the Semantic Web (URSW'05), collocated with the the 4th International Semantic Web Conference (ISWC'05)*, pp. 98–99, (2005).

[5] W. Hu and Y. Qu, 'Block matching for ontologies', in *the Proceedings of the 5th International Semantic Web Conference (ISWC'06)*, pp. 300–313, (2006).

[6] W. Hu, y. Zhao, and y. Qu, 'Partition-based block matching of large class hierarchies', in *the Proceedings of the 1st Asian Semantic Web Conference (ASWC'06)*, pp. 72–83, (2006).

[7] D. Lin, 'An information-theoretic definition of similarity', in *the Proceedings of the 15th International Conference on Machine Learning (ICML'98)*, pp. 296–304, (1998).

[8] M. Nagy, M. Vargas-Vera, and E. Motta, 'Dssim-ontology mapping with uncertainty', in *the International Workshop on Ontology Matching (OM'06), collocated with the 5th International Semantic Web Conference (ISWC'06)*, (2006).

[9] R. Pan, Z. Ding, Y. Yu, and Y. Peng, 'A bayesian network approach to ontology mapping', in *the Proceedings of the 4th International Semantic Web Conference (ISWC'05)*, pp. 563–577, (2005).

[10] E. Rahm and P.A. Bernstein, 'A survey of approaches to automatic schema matching', *Journal of VLDB*, **10**, 334–350, (2001).

[11] P. Shvaiko and J. Euzenat, 'A survey of schema-based matching approaches', *Journal of Data Semantics*, **4**, 146–171, (2005).

[12] Y. Wang, W. Liu, and D. Bell, 'Combining uncertain outputs from multiple ontology matchers', in *the Proceedings of the 1st International Conference on Scalable Uncertainty Management (SUM'07)*, pp. 201–214, (2007).

[13] Y. Zhao, X. Wang, and W.A. Halang, 'Ontology mapping based on rough formal concept analysis', in *the Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW'06)*, p. 180, (2006).

Ontology Evolution: A Practical Approach

Fouad Zablith*

Knowledge Media Institute (KMi), The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
f.zablith@open.ac.uk

Ontology evolution is increasingly getting research momentum in the Semantic Web field. This is due to the fact that ontologies, forming the backbone of Semantic Web systems, need to be kept up-to-date for ontology-based systems to remain usable. We highlight two research approaches in the domain of ontology evolution: The first considers the evolution as a pure management of changes performed by the user [7, 9–11], while the second takes into account dynamically updating and learning ontologies without offering extensive change and evolution management functionalities [1, 2, 8]. Many definitions of ontology evolution exist [5]. We understand ontology evolution as the "timely adaptation of an ontology to the arisen changes and the consistent management of these changes" [6]. This definition indirectly reflects the need of combining the two aforementioned approaches for achieving a successful evolution. Yet no practical and complete solutions exist that cover all stages of evolution.

We are planning to close the above gap by proposing a complete ontology evolution framework, Evolva[1] that: firstly covers the entire evolution cycle, and secondly makes use of background knowledge to potentially decrease, or even eliminate, user involvement. The need for Evolva emerged from the tedious and time consuming update and evolution of our KMi Semantic Web portal[2] ontology. Being highly user dependent and occurring in a dynamic domain, the ontology was left outdated. In this abstract we focus on the implementation of Evolva as part of the NeOn Toolkit[3], a novel ontology management framework. Figure 1 illustrates a screenshot of Evolva's pilot system.

Evolva detects the need for evolution by contrasting the content of the ontology to evolve (i.e. base ontology appearing in the left panel of Figure 1), with the content of external data sources. Such data sources can consist of text documents, databases, folksonomies, or even other ontologies, and can be selected in the "Data Sources" panel. Evolva processes the sources in its information discovery component in order to extract ontological entities. Currently we focus on concepts, but will extend the system to deal with instances as well. The Text2Onto [4] extraction algorithms are used for processing text documents and identifying entities. The entities are then passed to the data validation step that selects new entities with respect to the base ontology by using a Jaro-based string matcher. During validation, automated methods remove noisy terms that, for ex-

* This work is funded by the NeOn project sponsored under EC grant IST-FF6-027595
[1] An overview of Evolva can be found in [12] and [13].
[2] http://semanticweb.kmi.open.ac.uk/
[3] http://www.neon-toolkit.org/

ample, fall below a minimum term length threshold. We also have a filter for removing irrelevant terms such as the generic ones (e.g. thing, individual). The user is able to interfere in the validation, and manually exclude entities he/she believes are irrelevant to the domain. This is done under the "Data Validation" panel.

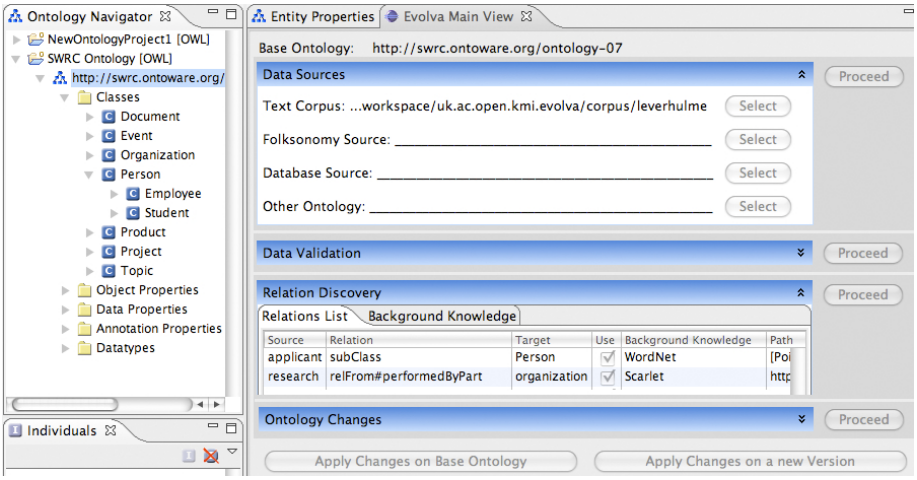


Fig. 1. Evolva Pilot System Screenshot

After the information discovery and validation stages, background knowledge is used for linking new and relevant entities to the base ontology. This is one of the core features of Evolva, as this stage is traditionally the most expensive in terms of user input. Background knowledge can be provided by different resources such as lexical databases, online ontologies and online documents. Our current implementation uses WordNet and online ontologies for relation discovery. WordNet contains hierarchy-based relations between terms and can be accessed quickly. Online ontologies are slower to access, but they offer a richer source of relations from a constantly increasing body of knowledge. We performed an experiment about the potential usage of such background knowledge sources for relation discovery, and they proved to have a high precision of around 77% [13]. Online ontologies are exploited using Scarlet[4], a relation discovery tool on the Semantic Web, from which hierarchy as well as named relations can be discovered. The "Relation Discovery" panel displays the *Source*, which is the new term extracted from the data sources, and its *Relation* to the *Target* term of the base ontology. Details of the relations such as the *Background Knowledge* used to discover it and its complete *Path* are also available. Figure 1 shows an example of how WordNet helped linking the new concept *Applicant* as a *subClassOf* *Person* (a concept in the base ontology). A second example shows how Scarlet

[4] http://scarlet.open.ac.uk/

links *Research* to *Organization*, through a *performedByPart* relation. The challenge here is to efficiently validate the relations, prior to applying any changes on the base ontology. E.g. how to select the right synset in WordNet, or how to determine whether a relation discovered from online ontologies does not conflict with the existing knowledge of the base ontology? Currently we are relying on the web-based distance similarity measure [3] as a step to check the possibility of two terms being related, before performing relation discovery. Part of our future plans is to use other validation techniques such as the base ontology itself as a validator, as well as word sense disambiguation. In addition to these automated validation methods, the user can manually exclude irrelevant relations.

The next step is to apply the changes on the base ontology using the relevant discovered relations. The changes can be applied either directly on the base ontology, or on a new detached copy of the base ontology. Our future implementation phase focuses on the two remaining components of Evolva: (1) evolution validation for consistency and duplication checks that could have occurred as an evolution side effect, and (2) the evolution management for recording changes and handling change propagation to the dependent components such as applications, or imported and aligned ontologies.

References

1. H. Alani, S. Harris, and B. O'Neil. Winnowing ontologies based on application use. *Proceedings of ESWC*, 2006.
2. Stephan Bloehdorn, Peter Haase, York Sure, and Johanna Voelker. *Ontology Evolution*, pages 51–70. John Wiley & Sons, June 2006.
3. R. L. Cilibrasi and P. M. B. Vitányi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, pages 370–383, 2007.
4. P. Cimiano and J. Völker. Text2onto - a framework for ontology learning and data-driven change discovery. *Proceedings of NLDB'05*, pages 15–17, 2005.
5. G. Flouris, D. Manakanatas, H. Kondylakis, D. Plexousakis, and G. Antoniou. Ontology change: classification and survey. *The Knowledge Engineering Review*, 23(02):117–152, 2008.
6. P. Haase and L. Stojanovic. Consistent evolution of owl ontologies. *Proceedings of ESWC*, pages 182–197, 2005.
7. M. Klein. *Change Management for Distributed Ontologies*. PhD thesis, Vrije Universiteit in Amsterdam, 2004.
8. V. Novacek, L. Laera, and S. Handschuh. Semi-automatic integration of learned ontologies into a collaborative framework. *IWOD*, 2007.
9. N. F. Noy, A. Chugh, W. Liu, and M. A. Musen. A framework for ontology evolution in collaborative environments. *Proc. of ISWC'06*, pages 544–558, 2006.
10. L. Stojanovic. *Methods and Tools for Ontology Evolution*. PhD thesis, FZI - Research Center for Information Technologies at the University of Karslruhe, 2004.
11. D. Vrandecic, H. S. Pinto, Y. Sure, and C. Tempich. The diligent knowledge processes. *Journal of Knowledge Management*, 9:85–96, 2005.
12. F. Zablith. Dynamic ontology evolution. *ISWC Doctoral Consortium*, 2008.
13. F. Zablith, M. Sabou, M. d'Aquin, and E. Motta. Using background knowledge for ontology evolution. *IWOD*, 2008.

Automated Access Control Rule Generation via Semantic Matching

Rui Zhang[1]

Abstract. Semantic Web techniques bring us help in many fields. In this paper, we propose a way to use Semantic Matching on access control. We illustrate the motivation with an eBusiness access control schema based on $RelBAC$ (for Relation Based Access Control). Semantic Matching techniques are applied on the lightweight ontologies of the subjects and the objects to find the semantic similarities that can be used to suggest new rules, to reuse the existing rules or to separate the duties of semantically disjoint user sets.

1 Introduction

Information Era releases people and data from centralized local environment to dynamic evolving communities and distributed information resources of various scales and types. $RelBAC$ [8] has been proposed as a new model for the dynamic evolving community access control scenario such as eBusiness. One important feature of $RelBAC$ is that hierarchies are naturally represented with a partial order '$\geq$' which is formalized as subsumption in the logical framework of $RelBAC$, i.e., an access control domain specific description logic. OWL-DL can be used to represent the knowledge as an ontology. This brings us not only the expressiveness, but also the possibility of applying semantic web techniques on the model. Meanwhile, dynamic community access control requires powerful management and administration on various scaled information. To generate new rules on the fly for this vast amount of changes will be time-consuming and error-prone. Thus suggestions to create new rules or to reuse existing rules arouse the interests of research.

We present in this paper a new way of applying Semantic Matching techniques [6] on access control. With a running example of an eBusiness schema, we show how to use these matching results to find semantically related subjects and object in order to suggest possible permission assignment; and to find the similar subject/object sets that can reuse existing rules. Even the mismatch between subject/object ontologies are useful such as for separation of duties.

The paper is organized as follows: Sec.2 describes an eBusiness access control schema based on $RelBAC$ as the motivation; Sec.3 shows how to apply Semantic Matching on access control; Sec.4 lists the state of the art and we conclude in Sec.5.

2 Motivation: eBusiness via $RelBAC$

Nowadays, eBusiness becomes so popular that the person sitting beside you on a trolley bus might be an eBusiness vendor of several online shops. Here we suppose an example in an eBusiness solution. An online vendor, Alice, has a shop on eBay selling digital devices. Her social network consists of many persons, e.g., Bob and David

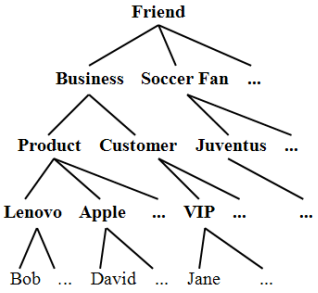


Figure 1. Alice's Social Ontology

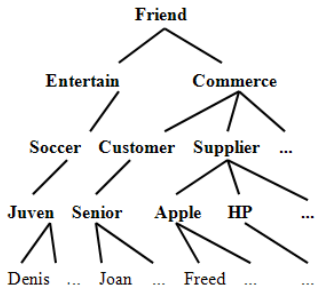


Figure 2. Bob's Social Ontology

have business relations with her and Chris and George are just common friends, etc. With the continuous growth of this network, Alice wants to manage these friends in her own way, so that she can easily find the 'proper' profile of a friend whenever necessary. For instance, David is a business friend who works as the sales of Apple company, and he will inform Alice the news of Apple products and special offers such that Alice can put it on her website in time. Jane is a representative of the best customer because she visits Alice's online shop frequently and comments on the deals she completed such that potential customers will get an impression on the quality of service and goods. Of course, Alice is happy to give Jane VIP prices as rewards. In general, Alice has a social network with various people and different social interactions.

As an eBusiness runner, Alice likes and has to control the access to the data she puts online. A natural and flexible access control model is $RelBAC$. As described in [8] $RelBAC$ is a model for community access control. It has common components such as `SUBJECT`s and `OBJECT`s, and a special part `PERMISSION`s as binary relations. A `PERMISSION` is a named pair $P(s, o)$ where s is a `SUBJECT`, o is an `OBJECT` and P is the `PERMISSION` describing the action that u intends to perform over o such as $Read$ and $Write$. $RelBAC$ defines a common relation with partial order '$\geq$' such that all these three components can be organized in hierarchies as a tree (or DAG). An access control domain-specific Description Logic is used to formalize the $RelBAC$ model. `SUBJECT`s and `OBJECT`s are formalized as concepts, and `PERMISSION`s as roles. Hierarchies in the model can be formalized as subsumption axioms. All the system states and access control policies are formalized as logical formulas on which automated reasoning can be performed.

Alice may build a tree-like structure as Figure 1 to classify people in her complex social network according to the social relations. The access control is simplified as managing the links between the sub-

[1] DISI, University of Trento, Italy, email: zhang@disi.unitn.it

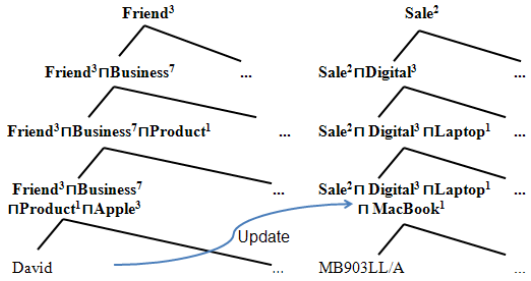


Figure 3. Permission Assignment in $RelBAC$

ject and object ontologies. By exploiting the theory of Lightweight Ontology as described in [5, 7], the arbitrarily manual structure is transformed into a lightweight ontology where implicit semantics on the tree edges are unified into explicit 'IS-A' relations, and the natural language labels of nodes are disambiguated with natural language processing [12] into logical formulas. For instance, Figure 3 shows parts of the lightweight ontologies built by Alice and the assignment of 'Update' to user 'David' on the set of objects 'MacBook'. In the left lightweight ontology, David is classified as an instance of the set '$Friend^3 \sqcap Business^7 \sqcap Product^1 \sqcap Apple^3$' according to his social position that he has a $Business^7$ relation with Alice and he works for $Apple^3$ (the superscript depicts the 3rd sense in the knowledge base, i.e., an IT company rather than a fruit). Symmetrically, in the right ontology (of the goods on sale), there is a class of objects '$Sale^2 \sqcap Digital^3 \sqcap Laptop^1 \sqcap MacBook^1$' where $Sale^2$ is a branch of $Business^7$, $MacBook^1$ is a $Laptop^1$ as a $Product^1$ of $Apple^3$. Apparently the two concepts are syntactically different, but semantically overlapping.

Things become more complicated when new ontologies arrive, e.g, if Alice likes to collaborate with some other eBusiness vendor who has her own user community and product category ontologies heterogeneously. The traditional way to solve the heterogeneity is to merge the database and create new rules for the 'new' knowledge base. Figure 1 shows part of Alice's social ontology and Bob, another eBusiness vendor has his own social ontology as Figure 2. The collaboration of Alice and Bob might lead to integration of these social 'resource's, such as product supplier, transporter, customer, etc. in addition to the integration of the physical resources such as goods.

So the motivation lies in at least two aspects:

- Semantic similarities disclose the latent relationships between subjects and objects although they are syntactically different. These latent relationships might suggest rules to be created for these semantically relevant subjects and objects such as to permit David to update the web categories about Apple products.
- Semantic similarities between ontologies of a type, such as between two subject ontologies or two object ontologies or even permission ontologies, provide a way to reuse (i.e., propagate) the permissions assigned by existing rules such as to reuse the rules for 'VIP' users of Alice onto Bob's 'Senior' customers.

3 Semantic Matching for Access Control

$RelBAC$ provides automated reasoning about the knowledge base such as consistency checking and query answering. Thus, *membership checking, security property enforcement* are used at design time to reason about *hierarchy management, permission propagation, separation of duties*, etc. and *query answering* can be used at run time for *access control decision*. However, that is not enough as an access control system for the larger and more complex eBusiness solutions crucially needs help to manage access control rules such as addressed in Section 2 by providing suggestions about candidate rules when the user is not an expert in access control (as it is often the case with social networks), or to provide semantic heterogeneity resolution for relatively large and complex ontologies or highly dynamic policies.

The fact that we handle subject, object and permission hierarchies as lightweight ontologies allows us to deal with the problem of semantic heterogeneity, namely with the fact that in general we will have multiple subject and/or object and/or permission hierarchies which express semantically related notions in many different forms. We can find with Semantic Matching tools that there exists similarity between the subject and object lightweight ontologies although they are heterogeneous and built independently. This will help to generate candidate permissions to be submitted to the user for approval, or generate semantically motivated constraints between subject and object categories, and so on.

To detect these semantic relations between classifications we use S-Match, a Semantic Matching tool described in [6]. The original idea of Semantic Matching is to calculate the semantic similarity such as *equal, overlapping*, etc. between the categories of the two given classifications. The core of a S-Match procedure consists two rounds of matching. The first round match is performed on the *concept at label* which are logical formulas formed with word senses such as the column names of Table 1. WordNet [9] is used as a knowledge base in which possible relations between senses (meanings of word) are provided. Semantic similarities are defined with sense relations. *Equal* $\equiv$: one concept is equal to another if there is at least one sense of the first concept, which is a synonym of the second. *Overlapping* $\sqcap$: one concept is overlapped with the other if there are some senses in common. *Mismatch* $\perp$: two concepts are mismatched if they have no sense in common. *More general / specific* $\sqsupseteq, \sqsubseteq$: One concept is more general than the other iff there exists at least one sense of the first concept that has a sense of the other as a hyponym or as a meronym. These direct results from the knowledge base can be regarded as a preparation for the second round of matching as they discover the relations between senses of single nodes. Afterwards, matching is performed on the *concept at node* which is a *conjunction* of all the *concepts at label* of nodes from the root to current, e.g., DL formulas in Figure 4. The results of the second round match is calculated by checking subsumption with a reasoner.

Let us see how to use these matching results in turn.

3.1 Suggestions for Rule Creation

For any access control systems, the stage of rules creation is very important because a cute rule set will simplify later work as enforcement and management. Semantic Matching between the subject and the object ontologies will find out potential semantic relations between categories of the two ontologies. For example, given the background knowledge about the relations $MacBook^1$ is a $Laptop^1$ as a $Product^1$ of $Apple^3$ etc., we can find the semantic similarities as listed in Table 1. As WordNet does not 'know' the word such as 'MacBook', which is common under the enormous emergences of new words in this Information Era, we should enrich the knowledge base with the facts such as '$Apple^3$ *is a IT company selling digital products such as MacBook and IPod.*'. This is a non-trivial task and many domain experts together with volunteers like common web

Table 1. Semantic Matching on Labels

S-Match	$Friend^3$	$Business^7$	$Product^1$	$Apple^3$	$Lenovo^1$	$Soccer^1 \sqcap Fan^2$
$Sale^2$	$\bot$	$\sqsubseteq$				$\bot$
$Digital^3$						
$Laptop^1$			$\sqsubseteq$			
$MacBook^1$				$\sqcap$	$\bot$	
$Thinkpad^1$				$\bot$	$\sqcap$	

users are contributing in this direction, at least to our own knowledge bases.

From Table 1, we can see the semantic similarities such as $Sale^2 \sqsubseteq Business^7$, etc. These relations provide the following suggestions to create new rules.

Semantically Related The cells marked with '$\sqsubseteq, \sqsupseteq, \equiv, \sqcap$' represent the semantic similarity of the corresponding concepts. It is meaningful to assign corresponding users some access to the objects. For example, the relation $Sale^2 \sqsubseteq Business^7$ suggests that some access, let us say $Read$, should be assigned to the $Business^7$ $Friend^3$ to some $Sale^2$ categories. It is obvious here in the small toy user and object ontologies, but facing a large eBusiness such as Amazon.com, these similarities will be very useful for the administrators in creating new rules. We may also place degrees on similarities. '$\equiv$' weighs more than '$\sqsubseteq$' and '$\sqsupseteq$', which in turn more than '$\sqcap$'. Therefore, it is more likely to assign access between '$\equiv$' related subjects and objects than the others.

Explicit Unrelated The cells marked with '$\bot$' represent that the corresponding concepts are found 'unrelated' in the knowledge base. Here we shorten the axiom '$C_1 \sqcap C_2 \sqsubseteq \bot$' as '$C_1 \bot C_2$'. We have to differentiate the real world semantics of these '$\bot$'s.

- $Sale^2 \bot Friend^3$ is a mismatch because they are referring to object and subject, i.e. an activity and a person respectively. This mismatch comes from the disjointness between person and activity as different subjects but does not prevent that a person can have some relation with an activity such as $Friend^3$ may have access to $Sale^2$.
- $MacBook^1 \bot Lenovo^1$ comes from that '*MacBook is a product of Apple company but not Lenovo.*' This kind of mismatch suggests exactly no access should be assigned.
- $Sale^2 \bot (Soccer^1 \sqcap Fan^2)$ covers both upper cases so it does prevent the access assignment from $Soccer^1$ Fan^2 to $Sale^2$.

The second case is a *strict mismatch* which means 'irrelevant' in common sense. It is important to detect this kind of mismatches because they can suggest for constraints such as *separation of duties* that we will discuss when matching two subject ontologies in the next subsection.

Implicit Unrelated The blank cells of the table mean that the knowledge base doesn't know any existing relation between the corresponding concepts. In this case, no semantical similarities are provided. From Table 1 we can see that this kind of cells are the majority in this example, only because the knowledge base we use is not designed for eBusiness domain. If it is specially enriched with more background knowledge, we believe more semantic relations can be found and more suggestions will be provided.

The interesting thing here is that the relation between $Friend^3$ and $Sale^2$ is *mismatch*. It is weird but true as $Friend^3$ means '*a person with whom you are acquainted*' and $Sale^2$ is '*the general activity of selling*'. This is common when we match a subject ontology with an object ontology. If we went on with the second round of S-Match on the *concepts at node* which includes all the semantics from the root to the current node, this mismatch between $Friend^3$ and $Sale^2$ would propagate to all the results and Table 1 would be full of '$\bot$s' simply because of the similarity of the two roots is *mismatch*. We may get nothing from such a table, therefore in this phase, we use only the first round of S-Match on *concepts at label*.

3.2 Automated Rule Reuse

One important evolution of subject and object ontologies is to integrate other similar ontologies. For example, an eBusiness vendor will enlarge her social network to involve more customers and very likely she would integrate the customer ontology of another vendor, or symmetrically integrate the goods ontology. The traditional access control solutions ask an administrator to create new rules for these evolving parts. Even for the similar ontologies, all assignments have to be made once again. For example, the vendor in the scenario of Section 2 would like to merge another ontology of subjects as Bob's Social Ontology as Figure 2. In this case for instance, a customer set called 'Senior' has the similar intuition to the 'VIP' set in previous ontology.

The resulting semantic relations can be used along the lines of what described in the previous sections either to drive the merging of the two ontologies or to create mappings which allow for the propagation of permissions from one ontology to the other. Thus for instance the system administrator might enforce that the equivalence mapping between the two root nodes in Figure 4 means that a Read permission on the left root node propagates to the right root node. These kinds of mappings are very similar to the C-OWL mappings introduced in [1] and should be used whenever a full merge of the two ontologies is not advisable or there are good reasons to keep the two ontologies distinct.

We show in Figure 4 the results of S-Match on two branches of the 'friend' lightweight ontologies generated from the hierarchies in Figures 1 and 2. The semantic similarity axioms can be added to the knowledge base of access control and the rule reuse is done without further efforts. For example,

$$\{(Friend^3 \sqcap Commerce^1) \sqsubseteq (Friend^3 \sqcap Business^7),$$
$$Business^7 \sqsubseteq \alpha\} \models Friend^3 \sqcap Commerce^1 \sqsubseteq \alpha$$

With the help of these semantic similarities found by S-Match, any *subject-centric* rules with permissions assigned to $Business^7$ will also propagate to $Friend^3 \sqcap Commerce^1$ just as a reasoning result without creating new rules for the new subject sets. Similar reuse applies on objects as well when S-Match is used to find the semantic similarities between object ontologies.

Even though indicating 'explicit unrelated', '$\bot$' is an important semantic similarity for rule reuse. Here we refer to the *strict mismatch* discussed in Section 3.1. It means that the two nodes in the two ontologies matched are semantically mutual exclusive, e.g.,

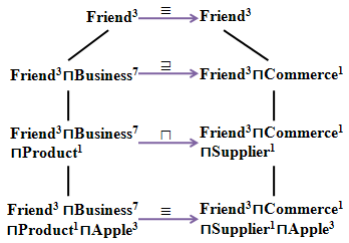


Figure 4. Ontology Matching for Rule Reuse

$HP^2 \perp Lenovo^1$ between the two ontologies in Figures 1 and 2. HP^2 and $Lenovo^1$ represent the set of users belonging to different IT companies, and it is rational to separate the duties from the two sets, i.e. users from company HP^2 should not have the same access as those from $Lenovo^1$. When the two ontologies are both considered as subject knowledge, the matching results suggest a new policy as $HP^2 \sqcap Lenovo^1 \sqsubseteq \perp$ which ensures that users cannot be members of both sets.

4 Related Work

With the arrival of Web 2.0 and now coming even Web 3.0, access control over the resources online throughout the evolving social networks demands more automated tools for administration.

Classic access control techniques, e.g., cryptography have been proposed for community access control such as [2]. However, this kind of access control systems focus on protection from security threats rather than taking use of the rich information from the web. The authentication procedure is done once for all which is not enough for fine-grained access control.

Lockr [11] was proposed to fit the situation that the large number of content sharing systems and sites use different access control methods un-reusable for each other. It separates social networking information from the content sharing mechanisms, so that end users do not have to maintain several site-specific copies of their social networks. It also provides a way to use social relationships as an important attribute, *relationship type*, to define access control rules. However, Lockr still uses a public/private key communication and does not consider the semantic similarities.

Another thread similar to our solution is Semantic Based Access Control. Yague et al. discussed the Semantic Access Control model in [3] with a XML based language SPL (Semantic Policy Language). The model is based on the semantic properties of the resources, clients (users), contexts and attribute certificates and relies on the rich expressiveness of the attributes to create and validate access control policies. It is flexible to define access control over attributes but faces the complexity problem of the system. In contrast, our model covers the expressiveness of attributes and takes use of the structure at the same time so that the permission propagation will greatly reduce the number of rules. Pan et al. present a novel middle-ware based system [10] to use semantics in access control. It is based on $RBAC$ model [4] with a mediator to translate the access request between organizations by replacing roles and objects with matched roles and matched objects. For interoperation, they use semantic mapping on roles in order to find the similarity or separation of duties between roles in two ontologies. This is similar to our approach, but we do much further as the S-Match tools are not domain specific so that we can match a subject ontology with an object ontology for new rule suggestions.

5 Conclusion

Based on the $RelBAC$ formalization of the access control problem in social networks, we can organize users, objects and permissions as (lightweight) ontologies. This allows to represent access control rules and policies as DL formulas and to reason about them using state of the art off-the-shelf reasoners. However, when the knowledge base is more and more complex, the rule management task explodes. Thus it requires automated or semi-automated tools to help creating and reusing rules. In this paper, we have shown how it is possible to use Semantic Matching technology to discover and exploit the underlying semantic relations between subject and object ontologies and between two user or object ontologies belonging to different policies. The resulting automated reasoning capabilities can be exploited to support the user or system administrator in the policy management, an activity which is time expensive and error-prone.

ACKNOWLEDGEMENTS

I would like to thank Prof. Fausto Giunchiglia and Bruno Crispo for collaboration on this work. Gratefulness should also be shown to all the KnowDive group members for suggestions and feedbacks.

REFERENCES

[1] Paolo Bouquet, Fausto Giunchiglia, Frank Van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt, 'C-owl: Contextualizing ontologies', in *Journal Of Web Semantics*, pp. 164–179. Springer Verlag, (2003).

[2] Barbara Carminati and Elena Ferrari, 'Privacy-aware collaborative access control in web-based social networks.', in *DBSec*, ed., Vijay Atluri, volume 5094 of *Lecture Notes in Computer Science*, pp. 81–96. Springer, (2008).

[3] Mariemma Inmaculada Yage del Valle, Mara del Mar Gallardo, and Antonio Mana, 'Semantic access control model: A formal specification', in *ESORICS*, eds., Sabrina De Capitani di Vimercati, Paul F. Syverson, and Dieter Gollmann, volume 3679 of *Lecture Notes in Computer Science*, pp. 24–43. Springer, (2005).

[4] David F. Ferraiolo, Ravi S. Sandhu, Serban I. Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli, 'Proposed NIST standard for role-based access control', *Information and System Security*, **4**(3), 224–274, (2001).

[5] Fausto Giunchiglia, Maurizio Marchese, and Ilya Zaihrayeu, 'Encoding classifications into lightweight ontologies.', *J. Data Semantics*, **8**, 57–81, (2007).

[6] Fausto Giunchiglia, Mikalai Yatskevich, and Pavel Shvaiko, 'Semantic matching: Algorithms and implementation.', *J. Data Semantics*, **9**, 1–38, (2007).

[7] Fausto Giunchiglia and Ilya Zaihrayeu, *Encyclopedia of Database Systems*, chapter Lightweight Ontologies, number 978-0-387-35544-3, Verlag, Springer, June 2009.

[8] Fausto Giunchiglia, Rui Zhang, and Bruno Crispo, 'Relbac: Relation based access control', in *International Conference on Semantics, Knowledge and Grid, SKG 2008*, ed., IEEE Computer Society, (2008).

[9] George A. Miller, 'Wordnet: A lexical database for english', *Communications of the ACM*, **38**, 39–41, (1995).

[10] Chi-Chun Pan, Prasenjit Mitra, and Peng Liu, 'Semantic access control for information interoperation', in *SACMAT '06: Proceedings of the eleventh ACM symposium on Access control models and technologies*, pp. 237–246, New York, NY, USA, (2006). ACM.

[11] Amin Tootoonchian, Kiran Kumar Gollu, Stefan Saroiu, Yashar Ganjali, and Alec Wolman, 'Lockr: social access control for web 2.0', in *WOSP '08: Proceedings of the first workshop on Online social networks*, pp. 43–48, New York, NY, USA, (2008). ACM.

[12] I. Zaihrayeu, L. Sun, F. Giunchiglia, W. Pan, Q. Ju, M. Chi, and X. Huang, 'From web directories to ontologies: Natural language processing challenges', in *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007), Busan, South Korea*, ed., Karl Aberer et al., volume 4825 of *LNCS*, pp. 617–630, Berlin, Heidelberg, (November 2007). Springer Verlag.

Evaluation of Ontology Mapping Representations

Hendrik Thomas[1], **Declan O'Sullivan**[1], **Rob Brennan** [1]

Abstract. A common approach to mitigate the effects of ontology heterogeneity is to discover and express the specific correspondences between different ontologies. An open research question is: how should such ontology mappings be represented. In recent years several proposals for an ontology mapping representation have been published, but till today no format is officially standardized or generally accepted in the community. In this paper we will present a new evaluation framework for ontology mapping representations for a pragmatic state of the art overview of their characteristics. In particular we are interested how current ontology mapping representations can support the management of ontology mappings (sharing, re-use, alteration) as well as how suitable they are for different mapping tasks.

1 INTRODUCTION

Ontologies are an important component for the implementation of the semantic web vision [1,2]. The promise of ontologies is to enable the sharing of a common understanding of a domain of interest that can be flexibly communicated between users and applications [3,4]. However, the actual conceptualization of a domain and the succeeding explication in an ontology language is a very heterogeneous process [5, 6]. For example, on a syntactical level a user can choose from a variety of ontology languages (e.g. RDF, OWL, Topic Maps, etc.) [5,7]. On a terminological level one can encounter all forms of mismatches related to the process of naming of ontology entities (e.g. synonymy, homonyms, multilanguage) [8]. Furthermore conceptual heterogeneity of ontologies arises due to the natural human diversity involved in modeling a domain [9,10], e.g. two ontologies could differ because they cover different (even overlapping) portions of the domain, provide a more (or less) detailed description or simply could reflect different viewpoints of the same domain. Finally, on a pragmatic level, one can encounter discrepancies related to the fact that different individuals may interpret the same ontology in different ways in different contexts [5,11]. Overall these levels of heterogeneities are major obstacles to the promised interoperability of ontologies [8].

A common approach to mitigate the effect of heterogeneity is to discover the specific correspondences between the different ontologies and to document these correspondences using an appropriate mapping expression [12, 13, 14]. In particular ontology mapping can be defined as the task of relating the vocabulary of two ontologies sharing the domain in such a way that the structure of ontological signatures and their intended interpretations are respected [15]. Despite the increasing tool support in the last years (e.g. MAFRA [16] COMA++ [17], Ontology Alignment API [9]) ontology mapping is still a challenging, complex and time-consuming process [9,12,15,18]. The different related issues in ontology mapping have been widely addressed in literature [5,12, 19].

One key aspect, which is still open to discussion, is the question: how should ontology mappings be explicit represented [9,19]? In this paper we define an ontology mapping representation as an explicit specification of the correspondence between ontologies to improve their interoperability. In recent years several proposals and recommendations for such an ontology mapping representation have been published, but till today no representation specific format is officially standardized or even generally accepted in the semantic web community [12, 20]. Thus an ontology engineer, when confronted with the need to merge or align multiple ontologies, has a choice between multiple currently available ontology mapping representations, each with their individual strengths and weaknesses for a specific mapping task.

Publications focusing on ontology mapping representations are relatively rare compared to the huge number focusing on other related questions, e.g. matching algorithms to identify mapping candidates (e.g. [21]). However, some previous studies on ontology mapping systems, in particular in [9,15,2,22,23], provide some insight. Most of these previous evaluations focus primarily on the technical capabilities of matching and mapping tools [20,24] and less on applicability of mappings representations for different mapping tasks [5, 18]. In addition, only sparse information has been published on the support of reusability and management of mappings, e.g. definitions of what meta-data types are supported. Finally, the evaluation processes as well as the criteria sets used have been heterogeneous, which makes it difficult to identify trends and improvements over time. In summary, a detailed evaluation framework as well as a comprehensive and up-to-date evaluation focusing on the capabilities of current ontology mapping representations is currently missing.

In this paper we will present a new evaluation framework for ontology representations used for a systematic analysis of ontology mapping formats that provides a state of the art overview of their characteristics. In particular we are interested how the ontology mapping representations can support the management of ontology mappings (sharing, re-use, alteration) as well as how suitable they are for different mapping tasks. The results of this evaluation will be of interest for understanding ontology mapping interoperability issues and also to support ontology engineers in choosing the most suitable mapping representation for their application.

2 EVALUATION FRAMEWORK

In this section we outline our evaluation methodology, set high-level goals for ontology mapping representations and finally decompose each high-level goal into specific metrics that can be evaluated.

[1] Knowledge & Data Engineering Group, School of Computer Science and Statistics, O'Reilly Institute, Trinity College Dublin, Ireland, Email: {Hendrik.Thomas,Declan.OSullivan,Rob.Brennan }@cs.tcd.ie

2.1 Methodology

To be able to compare and evaluate ontology mapping representations, first we need to define a set of evaluation criteria. Then these criteria can be consistently applied to any desired representations. To derive the criteria we will apply the Goal Question Metric (GQM) method, this is a tried and tested method for a structured and replicable evaluation of software products [25,26]. GQM provides a hierarchical structured procedure starting with goals (object and the issue to be measured) for each relevant evaluation dimension [25]. Each goal is refined into several questions, to break down the issue to characterize the object of measurement. Each question is then refined into metrics (objective, subjective) in order to answer it in a quantitative way. The result of the application of the GQM method is a replicable and detailed specification of a measurement system targeting a particular set of issues and a set of rules for the interpretation of the measurement data [25].

In the following sub-sections we describe an evaluation framework for ontology mapping representations derived using this method.

2.2 Goals for Ontology Mapping Representations

The first task in the development of an evaluation framework is the identification of a suitable set of goals for ontology mapping representations. Turning to the literature of ontology alignment and mapping it can be observed that instances of ontology mapping types can be quite heterogeneous, ranging from simple equivalences relations, mathematical conversions too complex structural mappings [12,27,28]. Therefore one of the most fundamental goals of ontology mapping representations is (G1) *the ability to express a mapping relation*. The second aspect we considered is that the construction of a specific ontology mapping can be complex and time-consuming. In fact, it could be more complex than the knowledge expressed in the ontologies itself [12,20]. Instead of creating the same or similar mappings repeatedly it is important to have a goal (G2*) to enable sharing and reuse of existing mappings* to reduce the effort involved in the creation of mappings [8]. Besides these aspects, an ontology mapping representation (G3) *should be computationally efficient to process* [8] in order to support the pragmatic concerns of implementing ontology interoperability solutions.

In the following subsections a set of questions is derived for each of our three goals that expose the different evaluation criteria used to characterize ontology mapping representations.

2.3 Goal 1: Ability to Express a Mapping Relation

Ontology mapping representation applicability can be considered from the viewpoint of expressiveness in terms of which operators and functions are supported to express the relevant ontology elements in correspondence and their individual alignments [20].

The first question in this context is therefore: (Q1) which kinds of ontology elements can be addressed so they can become the subjects of a mapping? This includes a single relevant ontology entity, an individual ontology fragment (e.g. specified by a search query) as well as the ontology as a whole. To simplify the expression of correspondences between ontologies it is important to know (Q2) which predefined relation types are supported, e.g. equability, incompatibility [29]. The specific set of supported relation types and the number of predefined relation types are indicators of the applicability of the representation. Also relevant is the extendibility in terms of: is it possible to add new transformations or relation types and still preserve the interpretability and processing ability of the representation in applications, e.g. by using an ontology language [12]. Specific knowledge, e.g. the date of birth of Tim Berners Lee, can be represented in quite different formats or conventions [10]. As a result, ontology mappings often have to deal with all kinds of conversions to enable interoperability [5,8]. It is therefore interesting to define (Q3) which functions are supported by the ontology mapping representation to express conversion mappings? This includes functions to manipulate numerical vales, text and dates.

Probably the most complex task for an ontology engineer is the handling of conceptual heterogeneity, because there is always more than one valid way to model a domain of interest [8,18], e.g. an address can be represented a single property or as a list of instances. From an abstract point of view this means ontologies could differ because different ontology elements and/or relations are used to express the same meaning [10]. Therefore it is relevant to ask (Q4) what functions an ontology mapping representation supports to express how relevant knowledge can be extracted and rearranged to make it interoperable (structural mapping). This involves adding or removing classes, instances, attributes (e.g. variant name in Topic Maps) and relations. It is also relevant if such a structural mapping is limited to a single representation language or not, e.g. can a mapping format express the mapping between RDF and Topic Maps which have different syntax and semantics [30, 31].RDF and Topic Maps which have a different syntax and semantics [30, 31].

Tab. 1 gives an overview of all deducted criteria for this goal.

Criteria	Type	Examples
Question 1: Which kind of ontology elements can be addressed?		
Single ontology element	yes|no	OWL class, property
Ontology fragment	yes|no	SPARQUL Query: SELECT ?x WHERE { ?x <http://vcard-rdf/3.0#FN> "John" }
Ontology as a whole	yes|no	http://kdeg.org/nembes.owl
Question 2: Which relations types are predefined?		
Amount predefined types	0..X	3
List of predefined types	list	equivalence, subsumption
Extensibility	yes|no	add a “neighbour” relation
Question 3: Which function for conversion mappings are supported?		
Numerical function	yes|no	add, subtract, multiply
String functions	yes|no	delete leading white spaces
Date functions	yes|no	2006/12/31 to 31/12/2006
Question 4: Which function for structural mappings are supported?		
Add / remove classes	yes|no	remove class town
Add / remove instances	yes|no	add instance Dubln
Add remove relation	yes|no	add Dublin is-part-of Ireland
Add remove attributes	yes|no	remove a variant name
Language specific	yes|no	OWL specific mapping

Table 1. Goal 1: Ability to Express a Mapping Relation

2.4 Goal 2: Enable Sharing and Reuse of Existing Mappings

To make a decision as to if and how a mapping can be reused or updated it is essential to understand how the mapping was created in the first place. An analysis of the life cycle of an individual ontology mapping is helpful to identify relevant decisions and information sources used, e.g. which matching algorithms

have been used to identify the mapping candidates [16, 12,32]. Meta-data documenting this lifecycle is essential to facilitate sharing and reuse of mappings. An ontology mapping representation should provide suitable placeholders to store and make this kind of information retrievable in a structured and predictable way [33].

Previously we have defined a mapping lifecycle [12]. The first stage of the ontology mapping lifecycle is the characterization phase which needs to be documented in the mapping representation and thus forms our first question of this goal (Q1). In the characterization phase the ontologies are analyzed with respect to their amenability for mapping. This involves the initial discovery of the ontologies; hence an ontology mapping representation needs to provide information to identify the ontologies which are the subject of the mapping like an identifier, path or an URL. However, ontologies may change over time and therefore additional ontology versioning information is useful to decide if a mapping is still appropriate [8,10]. Furthermore information on the format of the mapped ontologies are helpful to decide if an existing mapping is applicable in a different context, e.g. OWL DL or full [12]. Due to the syntactical heterogeneity many mapping tools require an initial transformation into an internal canonical format [17]. This has an impact on the supported ontology syntax and a mapping representation should include information on the canonical format used. Due to the terminological heterogeneity in this phase usually the content of the ontology is analyzed in order to characterize the nature of the terms used [8]. Descriptions of term construction rules or domain-specific thesauri/vocabularies used can influence the selection of an appropriate matching algorithm and should therefore be documented in the mapping representation [34]. In general, poor quality ontologies or divergent modeling approaches can make mapping attempts quite difficult or even impossible [35]. As a result, measures (qualitative and quantitative) of the ontology and the modeling approach applied are useful to understand the decisions made in the mapping process [12,36]. Another vital part of this life cycle phase is the decision whether matching should be attempted between ontologies. This decision can be influenced by organizational policies which govern the expenditure of resources [37]. If so, these policies should be documented because they are vital to understand future mappings.

One of the most important tasks in this phase is the identification of mapping candidates, either identified by manual selection or by an automated matching algorithm. If candidates have been manually selected, detailed information on this process (participants, time, context) as well as on the provenance of the data needs to be accessible. For example, if mappings are reused in a different organization, another role might be more appropriate for selection of mapping candidates [38]. Alternatively a wide range of matching algorithms can be applied, ranging from lexical to semantic model-based matching schemes [21]. The matching algorithm has a major impact on mapping creation and therefore it is essential to document the name as well as the specific configuration of the matching algorithms used [9]. Different matcher algorithm might be suitable for mapping task and therefore any information related to the matcher selection process is helpful, e.g. type of the matcher (string, language, constraint, linguistic, reuse, graph, taxonomy, model or combination based [21].

The second stage of the ontology mapping lifecycle (Q2) is the mapping phase which needs to be documented in the mapping representation [12]. The objective of this phase is generation of the information necessary for the execution of mappings as well as the creation of mappings that are relevant to the context of usage. As in the previous phase, it is necessary to check possible mappings against organization policies which need to be documented [12,37].

The determination of mappings by applications as well as humans from matching candidates is difficult and involves a certain level of uncertainty [8,17]. Suitable points of reference help to make the deduction process more predictable [12]. This includes pre-existing validated and trusted mappings or an explicit definition of the mapping context. Based on this information, a variety of strategies may be suitable for creating the mappings. For future reuse it is therefore important to know which specific strategy was applied [12]. It is also relevant to record any confidence value calculated or assigned to the mapping during the mapping or match generation processes.

Criteria	**Type**	**Examples**
Question 1: How is the characterization phase documented?		
Ontology identifiers	yes|no	string based matcher
Version information	yes|no	ontology version 1.5.4.
Ontology format(s)	yes|no	OWL lite, RDF(s)
Canonical format	yes|no	XML schema used in OISIN framework [12]
Terms used	yes|no	link to relevant thesauri
Ontology measures	yes|no	count of classes
Matching policies applied	yes|no	policy of organisation A
Type of matching creation	yes|no	automated or manual
Info on manual matching	yes|no	link to documentation
identifier of the used matcher	yes|no	model based matcher
Matcher configuration	yes|no	parameter
Matcher type	yes|no	linguistic based matcher
Question 2: How is the mapping phase documented?		
Matching policies applied	yes|no	policy of organisation A
Used pre-validated mappings	yes|no	A;creator = B;author
Mapping context	yes|no	specification of use-cases
Confidence level	yes|no	5 of 10
Mapping strategy	yes|no	OISIN framework [12]
Question 3: How is the management phase documented?		
Distribution system	yes|no	peer-to-peer network
Version information	yes|no	map version 1.2.3
Format information	yes|no	INRIA 1.0
Conflict/consistency check	yes|no	conflict mapA vs. mapB
Author information	yes|no	Hendrik Thomas
Date of creation	yes|no	19.12.2008 17:00
Authority for changes	yes|no	see http://onto.authority.ie
Dependencies	yes|no	mapping A depends on B
Change propagation method	yes|no	newsgroups announcement
Question 4: How is the interpretation of the meta-data supported?		
URI to identify entities	yes|no	http://cs.tcd.ie/onto/fname
Human-readable labels	yes|no	first Name
Documentation	List	source code, publications
Documentation URI	yes|no	http://cs.tcd.ie/onto/docu

Table 2. Goal 2: Enable Sharing & Reuse of Existing Mappings

The last phase of the mapping life-cycle (Q3) is the management phase which needs to be documented in the mapping representation. Any distributed system may be suitable for sharing mappings but the mapping representation should at least specify where to find the latest mapping sources as well as version information in order to keep track of mapping updates. Also any representation should explicitly specify its own format version, to support forward and backward compatibility. If mappings are used in a different contexts it is necessary to verify if they are consistent or in conflict with the existing mappings. A mapping representation could support this challenging task by providing a placeholder for relevant information, e.g. a suggested detection strategy. In addition existing mapping information can be altered or withdrawn, e.g. if they are erroneous [10,35]. A mapping representation should provide lifecycle information to support this, for example [12]: Who created the mapping and who has authorization to make changes. Which existing mappings are influenced by the proposed alteration? How will the change be propagated?

Another important issue in this context is (Q4) how is the interpretation of the meta-data supported by the mapping representation? Applications commonly use unique URIs for unambiguous identification of entities [2]. However, humans depend on human-readable labels as well as sufficient documentation (source code, tutorials, publications) which explain how specific meta-data should be interpreted. Similar to the subject indictor resources of Topic Maps [34,39] it is also useful that the URI of the meta-data field should refer to such an explanatory document to make the representation more self-explanatory to a human. Table 2 gives an overview of all deducted criteria for the second goal.

2.5 Goal 3: Computationally Efficient to Process

A first aspect is the (Q1) compatibility of the representation. It is thereby relevant whether the representation is implementation independent or is limited to a specific application. Also relevant is the question how easily the representation can be manipulated, e.g. by using a common syntax like RDF. The second aspect (Q2) are tools to support creation, sharing [40], management and visualization of mapping results and representations. Table 3 gives an overview of all deducted criteria for the third goal.

Criteria	Type	Examples
Question 1: How is the comparative is the representation?		
Implementation independent	yes|no	MAFRA format
Syntax	yes|no	XML, RDF, OWL
Question 1: What tool support is available?		
Creation & editing tools	List	Ontology Alignment API
Sharing tools	List	-
Management tools	List	COMA++
Mapping visualization tools	List	MAFRA

Table 3. Goal G3: Computationally Efficient to Process

2.6 Selection of Ontology Mapping Representations

In addition to the previous defined criteria the evaluation framework must also contain a set of rules defining how a specific evaluation should be conducted. The key question is: which ontology mappings representations should be included in the evaluation? Currently there are several non-ontology based (e.g. Text, XML) and ontology based (e.g. RDF, OWL [2]) languages used to express mappings [8]. The problem is that there is no consistent usage of these languages or formats. In fact many mapping tools use the same languages to express mapping results (e.g. RDF is very common) but in different ways and as a consequence they support different functions and operators to express mappings [8,12]. From a pragmatic point of view it is therefore not enough to evaluate a representation language like OWL in isolation. It is more important to understand how ontology mapping representations instances are supported by the individual ontology mapping tools.

3 SUMMARY AND OUTLOOK

In our evaluation we analyzed overall 13 different mapping and matching applications (see appendix for a complete list). The selection include historical relevant and established tools but also examples of leading up-to-date matching application [24]. For each of the 22 supported ontology mapping representation instances, 31 different evaluation parameters were determined. The evaluation was conducted in early 2009 by the authors in the Knowledge and Data Engineering Group, Trinity College (Dublin). The complete evaluation results are available online at: *https://www.cs.tcd.ie/~thomash/mapping_eva/home.php.*

The evaluation created a large amount of data and the upcoming workshop is a perfect opportunity to discuss our results with researchers and industry partners in order to identify issues and to develop a better understanding of the advantages and limitations of current mapping representations. Also we hope for feedback to optimize our current evaluation framework and suggestions for other ontology mapping systems which need be include into our next evaluation.

In conclusion, the previous remarkable efforts to support the creation of ontology mappings are just the first step. Further research is needed to develop a powerful mapping representation which is essential for the management, sharing and reuse of ontology mappings to even begin to support the flexible communication of a common understanding of a domain between users and applications a scale large enough to control the overall information glut [2].

ACKNOWLEDGEMENTS

This work is funded by the Irish Government as part of the Higher Educations Authority PRTLI Cycle 4 project NEMBES.

REFERENCES

[1] Berners-Lee, T., Hendler, J., Lassila, O., The *Semantic Web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities*, Scientific American Magazine, http://www.sciam.com/article.cfm?id=the-semantic-webUNDprint=true, 2001.

[2] Antoniou, G., Harmelen, F., *A Semantic Web Primer* (Cooperative Information Systems),The MIT Press, 2004.

[3] Gruber, T. A, *Transitional Approach to Portable Ontology Specifications*, Knowledge Acquisition, 5, pp. 199-220, 1993.

[4] Fensel, D., Ontologies *Silver Bullet for Knowledge Management and Electronic Commerce*, 2nd edition, Springer-Verlag, Berlin, 2003.

[5] Pepijn, R. S. V., Dean, M. J., Benchcapon, T. J. M. , Shave, M., *An analysis of ontological mismatches: Heterogeneity versus interoperability*, AAAI, Spring Symposium on Ontological Engineering, Stanford, USA, 1997.

[6] Oscar, C. *A declarative approach to ontology translation with knowledge preservation*, Volume 116 Frontiers in A.I., 2005.

[7] McGuinness, D. L., Harmelen, F. van (eds.) *OWL Web Ontology Language - Overview*, http://www.w3.org/TR /owl-features/, 2004.
[8] Bouquet, P., Ehrig, M., Euzenat, J. et al., D2.2.1 *Specification of a common framework for characterizing alignment*, http:// inrialpes.fr/exmo/cooperation/kweb/heterogeneity/deli/kweb-221.pdf, 2005.
[9] Euzenat, J., *An API for ontology alignment*, in Proceedings of the International Semantic Web Conference (ISWC 2004), LCNS 3298, pp. 698-712, Springer, Berlin, Germany, 2004.
[10] Garshol, L. M: *Towards a Methodology for Developing Topic Maps Ontologies*, in Maicher, L, Siegel, A, Garshol, L. M. (eds.): Leveraging the Semantics of Topic Maps - Second International Conference on Topic Map Research and Applications, TMRA 2006, Leipzig, Germany, October 11-12, 2006, Berlin Heidelberg New York, Springer, 2007, pp. 20-31.
[11] Miller, T., Thomas H., *Indices, Meaning and Topic Maps: Some Observations*, in Maicher , L., Siegel, A., Garshol, L. M. Leveraging the Semantics of Topic Maps - Second International Conference on Topic Map Research and Applications, TMRA 2006, Leipzig, Germany, October 11-12, 2006, Berlin: Springer, 2007, pp. 130-139.
[12] O'Sullivan, Wade, V., Lewis, D. *Understanding as We Roam*, in IEEE Internet Computing, 11, (2), 2007, p26 - 33 DOI: http://doi.ieeecomputersociety.org/ 10.1109/MIC.2007.50.
[13] Rahm, E., Bernstein, P. A., *A survey of approaches to automatic schema matching*. The VLDB Journal, 10(4):334–350, 2001.
[14] Hameed, A, Preece, A., Sleeman, D., *Ontology Reconcilliation, Handbook of ontologies*, in Stabb S. and Suder R. (eds) International Handbooks on Information Systems, Springer Verlag, Berlin, Germany, 2004, pp 31-250
[15] Kalfoglou, Y., Schorlemmer, M. *Ontology mapping: the state of the art*. in The Knowledge Engineering Review, 18(1):1–31, 2003.
[16] Maedche, A., Motik, B., Silva, N., Volz, R., *MAFRA – an ontology mapping framework in the context of the semantic web*, in proceedings of the 3th International Conference Onto1ogies and the Semantic Web:, Siguenza, Spain, 2002.
[17] Aumüller, D., Do, H., Maßmann, S.n Rahm, E.: *Schema and Ontology Matching with COMA++*. in: Proceedings. of the 2005 ACM SIGMOD Int. Conference on Management of *Data*. ACM Press, New York, NY, USA, 2005; pp. 906-908.
[18] Falconer, S., Storey, M.-A., *A cognitive support framework for ontology mapping*. in Processings of the 6th International Semantic Web Conference ISWC2007, http://iswc2007.semanticweb.org/ papers/113.pdf, 2007.
[19] Euzenat et al. D2.2.3: *State of the art on ontology alignment*, ftp://ftp.inrialpes.fr/pub/exmo/reports/kweb-223.pdf, 2004.
[20] Euzenat et al. D2.2.6: *Specification of the delivery alignment format*, 2006.
[21] Shvaiko, P., Euzenat J., *A Survey of Schema-based Matching Approaches*, in DIT Technical Report DIT-04-87, 2004.
[25] Basili, V. R., Caldiera, G., Rombach, H. D., *Goal Question Metric Approach*, ftp://ftp.cs.umd.edu/pub/sel/papers/ gqm.pdf, 2000.
[37] Beigi, M., Calo, S., Verma, D., *Policy Transformation Techniques in Policy-based Systems Management*, in: proceedings of IEEE Policy 2004, Yorktown, NY, June 004.
[36] Burton-Jones, A., Storey, V., Sugumaran V., Ahluwalia, P., *Assessing the Effectiveness of the DAML Ontologies for the Semantic Web*, NLDB 2003 Natural Language Processing and Information Systems, 8th International Conference on Applications of Natural Language to Information Systems, June 2003, Burg (Spreewald), Germany, 2003, pp 56-69.
[31] deBruijn, J., Foxvog, D., Zimmerman, K., *Ontology Mediation Patterns Library*, IST SEKT project deliverable, 4.3.1, 2005.
[40] Conroy C., *Wildflowr: P2P Sharing of Ontology Mappings*, M.Sc. Dissertation, Trinity College Dublin, May 2005.
[38] Conroy, C. *Towards Semantic Mapping for Casual Web Users*, in: proc. of the 7th International Semantic Web Conference (ISWC2008), Heidelberg, Springer, 2008 pp. 907-913.
[22] Hong-Hai, D., Melnik, S., Rahm E., *Comparison of schema matching evaluations*. in: proc. GI-Workshop "Web and Databases", Erfurt (DE), 2002. http://dol.unileipzig.de/pub/2002-28.
[27] Euzenat J., *An API for ontology alignment*, International Semantic Web Conference (ISWC 2004), LCNS 3298, pages 698-712, Springer, Berlin, Germany, 2004.
[33] Fugmann, R., *Subject Analysis and Indexing: Theoretical Foundation and Practical Advice*, Frankfurt a. M., Indeks, 1993.
[30] Garshol, L. M., *Living with topic maps and RDF: Topic maps, RDF, DAML, OIL, OWL, TMCL*. http://www. ontopia.net/topicmaps/materials/tmrdf.html, 2002.
[39] Garshol, L. M., Moore, G. ISO/IEC JTC1/SC34, *Information Technology - Document Description and Processing Languages*, http://www.isotopicmaps.org/sam/sam-model/, 2006.
[29] Giunchiglia, F., Shvaiko, P., *Semantic matching*, in proceedings of the IJCAI Workshop on ontologies and distributed systems, pages 193-146, Acalpulco, Mexico, 2003.
[28] Maedche A., Motik B., Silva N., Volz R., *MAFRA - A Mapping FRAmework for Distributed Ontologies in the Semantic Web*, in proc. of the workshop on knowledge transformation for the semantic web (KTSW 2002), ECAI 2002, pages 60-68, Lyon, France, 2002.
[24] Noy, N., *Semantic Integration: A Survey of Ontology-Based Approaches*, in Special Issue on Semantic Integration, SIGMOD Record, Volume 33, Issue 4, pages 65-70, December 2004.
[23] Parent, C,. Spaccapietra, D. *Database integration: the key to data interoperability*. The MIT Press, Cambridge (MA US), 2000.
[35] Smith, B., *Ontology and Information Systems*, Lecture Text, http://ontology.buffalo.edu/ontology(PIC).pdf, 2000.
[26] Solingen, R. van, Berghout, E., *The Goal/Question/Metric Method: a practical guide for quality improvement of software development*, The McGRAW-HILL Companies, London u.a., 1999.
[34] Thomas, H., Redmann, T., Markscheffel, B. *Controlled semantic tagging – how can topic maps support subject indexing in digital libraries?*, In: Shoniregun, C. A., Logvynovskiy, A.: Proceedings of the International Conference on Information Society (i-Society 2007),2007, pp. 346-352.
[32] Yang, K., Steele, R., *A Framework for Ontology Mapping for the Semantic Web*, Proceedings of the International Conference on Information Technology in Asia, http://www-staff.it.uts.edu.au/ ~kayang/download/AFOMSW.pdf, 2007.

APPENDIX A OVERVIEW OF EVALUATED APPLICATIONS

Application	Link
Alignment API	http://alignapi.gforge.inria.fr/
Anchor-PROMPT	http://protege.stanford.edu/plugins/prompt/prompt.html
COMA++	http://dbs.uni-leipzig.de/Research/coma.html
Context Matching Algorithm (CtxMatch)	http://dit.unitn.it/~zanobini/downloads.html
CROSI Mapping System (CMS)	http://www.aktors.org/crosi/
Falcon-AO	http://iws.seu.edu.cn/projects/matching/projects.jsp
Framework for Ontology Alignment and Mapping (FOAM)	http://www.aifb.uni-karlsruhe.de/WBS/meh/foam/
Lily	http://ontomappinglab.googlepages.com/lily.htm
MAFRA	http://mafra-toolkit.sourceforge.net
MapOnto	http://www.cs.toronto.edu/semanticweb/maponto/
OntoBuilder	http://iew3.technion.ac.il/OntoBuilder
Ontology Mapping Tool OMT)	http://www.wsmx.org/
Risk Minimization based Ontology Mapping (RiMOM)	http://keg.cs.tsinghua.edu.cn/project/RiMOM/

Towards instance coreference resolution in a multi-ontology environment

Andriy Nikolov[1], Victoria Uren[1], Enrico Motta[1] and Anne de Roeck[1]

1 INTRODUCTION

With the growing amount of semantic data published on the Web the problem of coreference resolution gains in importance. The linked data initiative provided guidelines for publishing RDF datasets and new datasets are constantly being made available. Such datasets often contain descriptions of the same real-world entities but use different URIs to refer to them. In order to utilize published data on a web scale it is essential to detect such situations and resolve coreferences.

In the Semantic Web community initially research effort was primarily concentrated on schema-level ontology alignment and many tools have been developed [1]. With the growing amount of published data instance-level integration issues also started to receive attention recently [2], [4]. These systems abstract from schema-level issues and focus on finding coreferent instances assuming their type and structure to be the same.

In our view there is still a gap concerning the study of the complete data integration workflow. On the one hand, schema alignment algorithms do not support the level of granularity necessary for data processing (e.g., applying different settings for individuals of different class). On the other hand, data-level integration tools assume schema-level issues to be resolved and do not consider implications of automated schema alignment. Our system KnoFuss was initially developed to perform integration of automatically extracted annotations structured according to a single common ontology. We extended it to operate in a multi-ontology environment and to utilise schema alignments produced by automatic ontology matching tools. Here we describe the resulting system workflow and first findings obtained during initial tests.

2 EXTENDING KNOFUSS ARCHITECTURE FOR MULTI-ONTOLOGY COREFERENCE RESOLUTION

KnoFuss architecture [6] implements a modular framework for semantic data fusion. The architecture focused on two main data fusion subtasks: coreference resolution (finding identical instances) and knowledge base updating (refining coreferencing results and resulting knowledge base taking into account ontological constraints and data conflicts). Algorithms performing fusion subtasks are represented as problem-solving methods [5]. Their capabilities (range of applicability and reliability of output) are formally defined using the fusion ontology.

Obviously, the behaviour of each algorithm differs depending on the task to which it is applied: reliability of name matching using string similarity differs when individuals belong to a generic class *foaf:Person* or a specific class *sweto:Computer_Science_Researcher,* the same string metrics cannot be applied when comparing paper titles and person names because the order of words in the name can differ, etc. These differences are represented using *application contexts*: bridges between a specific domain and a method. For a coreference resolution method the context may define more precise reliability estimation, narrowed range of applicability and extended set of relevant properties.

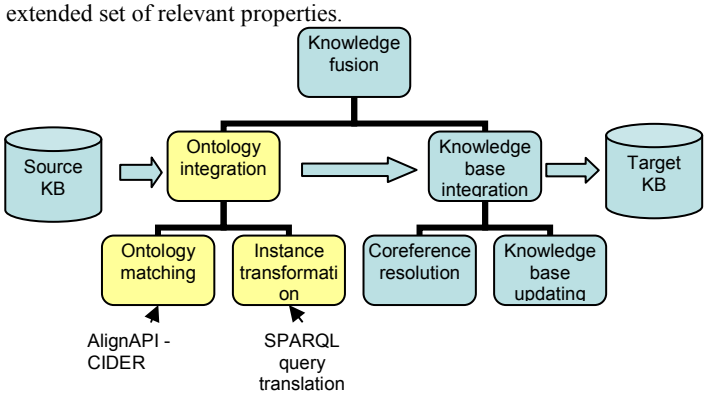


Figure 1. Fusion task decomposition in the KnoFuss architecture

Our ongoing work focuses on extending the functionality of KnoFuss to operate on a larger scale in a multi-ontology environment. If the source and target knowledge bases are structured according to different ontologies two additional subtasks are added to bridge the gap: *ontology matching* (obtaining schema alignments) and *instance transformation* (resolving structural differences between instances in two knowledge bases).

At the first step, schema-related statements are separated and available ontology matching algorithms are called to produce alignments (at the moment it is assumed that they produce their output in the standard AlignAPI format). The system utilizes two types of mappings: *equivalence* and *disjointness*. After candidate equivalence mappings are produced the system tries to generate additional disjointness relations:

- by inferring them using disjointness in a single ontology and available equivalence mappings;
- by querying background knowledge: the Scarlet service [7] is called to check whether disjointness between terms was defined in any other existing ontology on the Web.

In case of a logical conflict (e.g., when two classes are considered disjoint but their subclasses are equivalent), the conflict is resolved based on the similarity measure of corresponding mappings: less reliable mappings are excluded.

These automatically generated mappings are then used to perform instance transformation. In KnoFuss, applicability range and relevant attribute selection sets are defined as SPARQL queries in the terms of the target ontology. These queries are translated into the terms of the source ontology using available mappings. In the case when a term in the target ontology

potentially corresponds to several terms in the source ontology their union is considered: e.g.,

```
SELECT ?uri WHERE {
        ?uri rdf:type sweto:Computer_Science_Researcher }
is translated into
SELECT ?uri WHERE {
        {?uri rdf:type tap:CMU_Person}
UNION
        {?uri rdf:type tap:Computer_Scientist}
UNION
        {?uri rdf:type tap:Medical_Scientist}}
```

In this example there is a modelling style difference between two ontologies: individuals, which are classified as computer scientists in the SWETO ontology, are classified into several classes in TAP based on different criteria: place of work (*CMU_Person*) for some individuals and main research area (*Computer_Scientist* and *Medical_Scientist*) for others. Some authors of medical expert systems were assigned to the class *Medical_Scientist*. The ontology matching tool correctly identified these overlaps and produced three candidate mappings for the class *sweto:Computer_Science_Researcher*.

These pairs of queries assumed to be equivalent are then used at the later stages of the workflow, which allows the system to operate in the same way as in a single ontology case.

3 RESULTS AND DISCUSSION

We implemented a prototype of the system employing the CIDER tool [3] and performed initial tests trying to find coreferent individuals in two testbed knowledge bases: TAP and SWETO (Table 1). We applied different string metrics over individuals of several classes. Also we ran tests applying the CIDER ontology matching tool to measure instance similarity to compare its performance to standard string similarity metrics.

Some general initial findings are:

- As could be expected, errors during the schema matching stage are propagated and can potentially lead to significant distortions during instance coreferencing. For instance (rows 5 and 6), incorrect alignment of *tap:Country* to *sweto:Company* led to 30% precision drop (many companies have names derived from country names).
- Ontological constraints are extremely valuable in coreferencing task as a mean to repair such errors. Apart from the widely used *owl:FunctionalProperty* and *owl:InverseFunctionalProperty*, which allow non-ambiguous instance identification, *negative* evidence is also valuable for filtering out incorrect mappings. These constraints include disjointness and datatype properties with cardinality constraints. E.g., knowing that *Company* is disjoint with *Country* (or inferring that) would repair the problem in rows 5 and 6. Most ontologies do not define these explicitly, however, having a high-level reference ontology accessible on the Web where these constraints are specified would be a significant source of information.
- Label comparison cannot be considered sufficiently reliable evidence for coreference resolution. However, more complex algorithms utilizing context data (additional properties and links between individuals) can only be applied to datasets containing sufficiently overlapping data. It can be expected that many data integration tasks on the Web scale will only be able to rely on instance names and thus can only provide suggestions rather than generate *owl:sameAs* statements carrying strong implications.
- Although semantic heterogeneity (different meaning attached to similar resources) is primarily a schema-level knowledge modelling issue, it can cause problems on the instance level as well. For instance, the TAP ontology contains a single individual “Coca-Cola” while SWETO contains several individuals describing Coca-Cola branches in different countries. Whether such instances should be considered equivalent depends on the context of the task.
- Since errors are inevitable in automatic coreferencing, provenance information must be stored together with produced coreference mappings so that the user application can decide whether to rely on them or not. One possible way is to extend the coreference bundles approach [2] to include for each URI the confidence of its inclusion into the set.
- It is hard to find a single matching algorithm to apply to all kinds of data: settings have to be optimised for a specific type of data rather than for a specific pair of ontologies as in schema matching. For instance, optimal thresholds for CIDER differed significantly depending on the class (rows 2, 7 and 11).

REFERENCES

[1] Euzenat, J. and Shvaiko, P. Ontology Matching. Springer, 2007.

[2] Glaser, H., Millard, I., Jaffri, A., Lewy, T. and Dowling, B. On Coreference and The Semantic Web. In: *7th International Semantic Web Conference*, October 26 - 30, Karlsruhe, Germany. 2008 (Submitted)

[3] Gracia, J. and Mena, E., Ontology Matching with CIDER: Evaluation Report for the OAEI 2008, Proc. of 3rd Ontology Matching Workshop (OM'08), at 7th International Semantic Web Conference (ISWC'08), Karlsruhe (Germany), CEUR-WS, ISSN-1613-0073, October 2008.

[4] Yanbin Liu and Francois Scharffe and Chunguang Zhou. Towards Practical RDF Datasets Fusion. Workshop on data integration through semantic technology (DIST2008), Bangkok, Thailand, December 2008.

[5] E. Motta. Reusable Components for Knowledge Modelling. IOS Press, 1999.

[6] A. Nikolov, V. Uren, E. Motta, A. de Roeck. Integration of semantically annotated data by the KnoFuss architecture. EKAW, Acitrezza, Italy, 2008.

[7] Sabou, M., d'Aquin, M., Motta, E., Exploring the Semantic Web as Background Knowledge for Ontology Matching. Journal of Data Semantics XI, 2008.

N	Class (SWETO)	Algorithm	Precision	Recall	F1	Threshold
1	Person	L2 Jaro-Winkler	0.29	0.92	0.44	1.0
2	Person (2000 subset)	CIDER	0.37	0.95	0.53	0.05
3	Computer Science Researcher	L2 Jaro-Winkler	0.62	0.93	0.75	0.99
4	Organization	Monge-Elkan	0.42	0.95	0.58	0.93
5	Company	Monge-Elkan	0.41	0.95	0.57	0.92
6	Company (manual schema alignment)	Monge-Elkan	0.74	0.95	0.83	0.93
7	Company (2000 subset)	CIDER	0.93	0.64	0.76	0.14
8	Company (2000 subset)	Jaro-Winkler	0.79	0.74	0.77	0.92
9	City	Monge-Elkan	0.92	0.98	0.95	0.92
10	City (2000 subset)	Jaro-Winkler	0.80	0.99	0.89	0.92
11	City (2000 subset)	CIDER	0.91	0.61	0.73	0.28

Table 1. Subset of initial test results for instance coreferencing (TAP vs SWETO)