Extending HCONE-merge by approximating the intended meaning of ontology concepts iteratively

George A. Vouros, Konstantinos Kotis

Department of Information & Communications Systems Engineering,
University of the Aegean,
Karlovassi, Samos,
83200, Greece
{georgev, kkot}@aegean.gr

Abstract. A central aspect of HCONE-merge is the mapping of ontology concepts to a hidden intermediate ontology by uncovering the intended meaning of concepts. Such a mapping is realized by a semantic morphism from ontology concepts to WordNet senses. Extending methods that have already been proposed, this paper proposes an iterative algorithm for approximating the intended meanings of ontology concepts in a fully automated way. Results from numerous experiments are thoroughly described and conclusions are drawn.

1 Introduction

Ontologies have been realized as the key technology to shaping and exploiting information for the effective management of knowledge and for the evolution of the Semantic Web and its applications. Ontologies establish a common vocabulary for community members to interlink, combine, and communicate knowledge shaped through practice and interaction, binding the knowledge processes of creating, importing, capturing, retrieving, and using knowledge. However, it seems that there will always be more than one ontology even for the same domain [1]. In distributed settings, where different conceptualizations of the same domain exist, information services must effectively answer queries bridging the gaps between conceptualizations of the same domain. Towards this target, networks of semantically related information must be created at-request. Therefore, coordination (i.e. mapping, alignment, merging) of ontologies is a major challenge for bridging the gaps between agents (software and human) with different conceptualizations.

There are many works towards the mapping/merging of ontologies (e.g. [2] [3] [4] [5] [6]). These works exploit linguistic, structural, domain knowledge and matching heuristics. Recent approaches aim to exploit all types of knowledge and further capture the intended meanings of terms by means of heuristic rules [2]. The HCONE-merge approach to the merging of ontologies [7] [8] exploits linguistic, structural and semantic knowledge and gives much emphasis on "uncovering" the intended informal interpretations of concepts specified in an ontology. Linguistic and structural

knowledge about ontologies is exploited by the Latent Semantics Indexing method (LSI) [9] for associating concepts to their informal, human-oriented intended interpretations realized by WordNet senses. Using concepts' intended interpretations, the proposed mapping/merging method translates formal concept definitions to a common vocabulary and merges the translated definitions by means of description logics' reasoning services.

The HCONE-merge approach, as it was originally proposed, requires humans to validate the interpretations suggested by LSI for every term in the ontology. Since this process is quite frustrating and error-prone, even for small ontologies, we contact research towards minimizing the required human involvement for mapping concepts to their intended interpretations. The ultimate achievement would be to fully automate the mapping of concepts to their intended interpretations, and consequently to fully automate merging. Towards this goal, we have developed techniques and heuristics for ontology mapping and merging that require varying degrees of human involvement [8]. Although we managed to achieve a high degree of precision, this has been gained with the cost of considerable human involvement in the process.

Based on the HCONE method, the present paper proposes an iterative and automatic method for computing the mapping of concepts to their intended informal interpretations. This method requires no human involvement and, as experiments show, it converges to a set of mappings with high precision.

Section 2 of the paper provides definitions of notions used throughout the paper. Section 3 gives an overview of the HCONE-merge method and of research towards automating the computation of the mapping of ontology concepts to their intended meaning. Section 4 describes the iterative approximation of intended interpretations and section 5 provides results of experiments contacted. Section 6 concludes the paper.

2 Background Definitions

An ontology is considered to be a pair $O=(S, A)$, where S is the ontological signature describing the vocabulary (i.e. the terms that lexicalize concepts and relations between concepts) and A is a set of ontological axioms, restricting the intended interpretations of the terms included in the signature. In other words, A includes the formal definitions of concepts and relations that are lexicalized by natural language terms in S.

Ontology mapping from ontology $O_1 = (S_1, A_1)$ to $O_2 = (S_2, A_2)$ is considered to be a morphism $f:S_1 \rightarrow S_2$ of ontological signatures such that $A_2 \models f(A_1)$, i.e. all interpretations that satisfy O_2's axioms also satisfy O_1's translated axioms [3] [12]. Consider for instance the ontologies depicted in Fig. 1: Given the morphism f such that $f(Infrastructure)=Facility$ and $f(Transportation)=Transportation\ System$, it is true that $A_2 \models \{f(Transportation) \sqsubseteq f(Infrastructure)\}$, therefore f is a mapping. Given the morphism f', such that $f'(Infrastructure) = Transportation\ System$ and $f'(Transportation) = Transportation\ Means$, it is not true that $A_2 \models \{f'(Transportation) \sqsubseteq f'(Infrastructure)\}$, therefore f' is not a mapping.

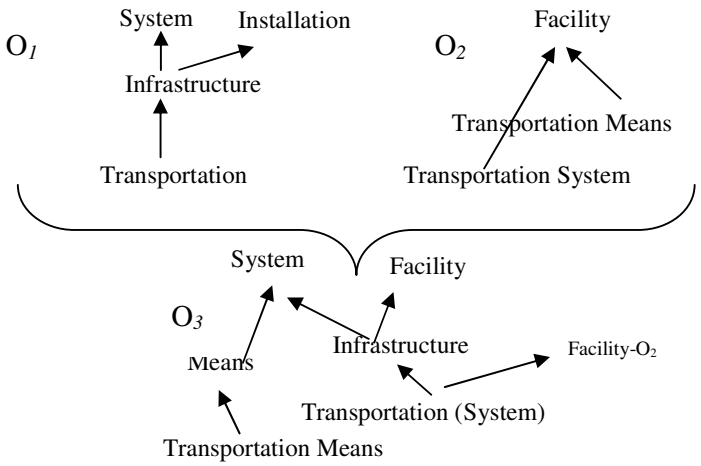


Fig. 1. Example Ontologies

However, instead of a function, we may articulate a set of binary relations between the ontological signatures. Such relations can be the inclusion ($\sqsubseteq$) and the equivalence ($\equiv$) relations. For instance, given the ontologies in Fig. 1, we can say that *Transportation*$\equiv$*Transportation System, Installation*$\equiv$*Facility* and *Infrastructure* $\sqsubseteq$ *Facility.* Then we have indicated an alignment of the two ontologies and we can merge them. Based on the alignment, the merged ontology will be ontology O_3 in Fig. 1. It holds that $A_3 \models A_2$ and $A_3 \models A_1$.

Looking at Fig. 1 in an other way, we can consider O_3 to be part of a larger intermediary ontology and define the alignment of ontologies O_1 and O_2 by means of morphisms f_1: $S_1 \rightarrow S_3$ and f_2: $S_2 \rightarrow S_3$. Then, the merging of the two ontologies is the minimal union of ontological vocabularies and axioms with respect to the intermediate ontology where ontologies have been mapped.

In the example of Fig.1, concepts *Transportation-*O_1 and *Transportation System-*O_2 will be found to have the same intended meaning, and therefore will be considered equivalent. The merging of their formal definitions will eventually result to:

Transportation System-$O_2 \sqsubseteq$ *Infrastructure-*$O_1 \sqcap$ *Facility-*O_2

However, the description logics classification mechanism will consider the axiom *Transportation System-*$O_2 \sqsubseteq$ *Facility-*O_2 to be redundant (see Fig. 1). Therefore O_3 *will eventually* contain only the axiom *Transportation System* $\sqsubseteq$ *Infrastructure*. Doing so, the merged ontology contains only the minimal set of axioms resulting from source ontologies mapping.

The ontologies merging problem (OMP) can be stated as follows: *Given two ontologies find an alignment between these two ontologies, and then, get the minimal union of their (translated) vocabularies and axioms with respect to their alignment.*

3 HCONE-merge

The HCONE-merge method finds a morphism between each of the two original ontologies and the so-called "hidden intermediate" ontology. As it is shown in Fig. 2, where the overall method is depicted, WordNet plays the role of an "intermediate". We consider that each sense in a WordNet synset describes a concept. WordNet senses are related among themselves via the inclusion (hyponym – hyperonym) relation. Terms that lexicalize the same concept (sense) are considered to be equivalent through the synonym relation.

Ontology concepts are being mapped to WordNet senses. This mapping indicates the informal intended interpretations of concepts and it is specified by the semantic morphism (*s-morphism*, symbolized by f_s). Using this mapping, HCONE-merge constructs the intermediate ontology that includes (a) a vocabulary with the lexicalizations of the specific senses of WordNet synsets corresponding to the ontologies' concepts, and (b) axioms that are the translated axioms of the original ontologies. Having specified the mappings to the hidden intermediate ontology, the translated ontologies are merged following merging actions such as rename, merge, and classify.

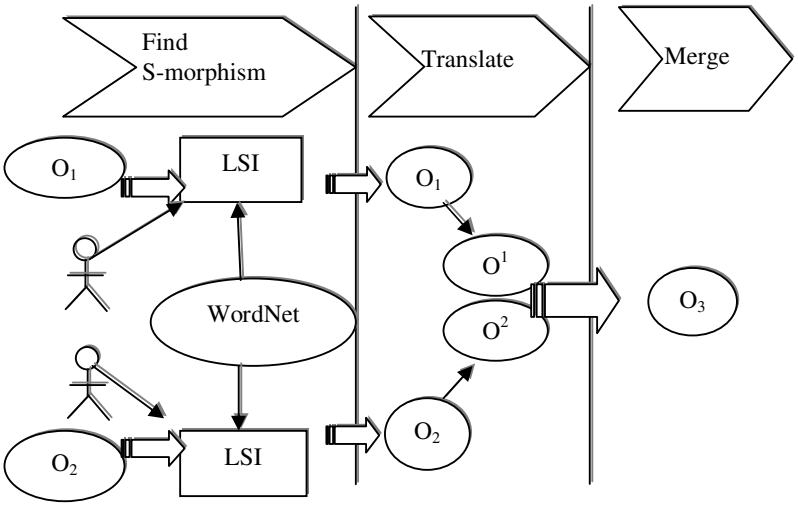


Fig. 2. The HCONE approach towards the OMP

It must be noticed that we do not consider WordNet to include any intermediate ontology, as this would be very restrictive for the specification of the original

ontologies (i.e. the method would work only for those ontologies that preserve the inclusion relations among WordNet senses).

The computation of the semantic morphism is based on the lexical semantic indexing (LSI) method.

LSI [9] is a vector space technique for information retrieval and indexing. It assumes that there is an underlying latent semantic structure that it estimates using a matrix of term-document association data by means of statistical techniques. In our case the $n \times m$ space comprises the n more frequently occurred terms of the m WordNet senses the algorithm focuses on. Lexical Semantic Analysis (LSA) allows the arrangement of the semantic space to reflect the major associative patterns in the data. As a result, terms that did not actually appear in a sense may still end up close to the sense, if this is consistent with the major patterns of association in the data. Position in the space then serves as the new kind of semantic indexing.

Given a query (which in our case corresponds to an ontology concept), retrieval aims to locate a point in space that is close to the sense that expresses the intended meaning of this concept. The query to the retrieval mechanism is constructed by the concept names and the associated senses of all concepts in the vicinity of the given concept. The steps of the algorithm for finding the semantic morphism are shown in Fig. 3.

1. Choose a concept from the ontology. Let *C* be the concept name.
2. Get all WordNet senses $S_1, S_2, \ldots S_m$, lexicalized by *C'*, where *C'* is a linguistic variation of *C*. These senses provide the *focus of the algorithm for* C.
3. Get the hyperonyms and hyponyms of all *C'* senses.
4. Build the "*semantic space*": An *n*X*m* matrix that comprises the *n* more frequently occurred terms in the *vicinity* of the *m* WordNet senses found in step 2.
5. Build a query string using the terms in the *vicinity* of *C*. The query string is a sequence of digits, each digit taking value 0 if a term in the *vicinity* of *C* does not exist in the set of *n*, and 1 if a query term exists in the set of *n*.
6. Find the ranked associations between *C* and *C'* senses by running the Latent Semantics Analysis (LSA) function and consider the association with the highest grade. LSA uses the query terms for constructing the query string and computes a point in the semantic space constructed in step (4).

Fig. 3. The algorithm for computing the s-morphism

The semantic space is constructed by terms in the vicinity of the senses $S_1, S_2, \ldots S_m$ that are in focus of the algorithm for a concept *C*. Therefore, we have to decide what constitutes the vicinity of a sense for the calculation of the semantic space. In an analogous way we have to decide what constitutes the vicinity of an ontology concept for the calculation of the query string.

Information that can be included in the semantic space includes:

- The term *C'* that corresponds to *C*. *C'* is a lexical entry in WordNet
- Terms that appear in *C'* WordNet senses
- Terms that constitute hyperonyms / hyponyms of each *C'* sense.

- Terms that appear in hyper(hyp)onyms of *C'* senses

Information that can be included in the query for a concept *C* includes:

- Concept's *C* primitive super-concepts.
- Concepts that are immediate super-concepts of *C*
- Concepts that are immediate sub-concepts of *C*
- Concepts that are related to *C* via domain specific relations
- The most frequent terms in WordNet senses that have been associated with the concepts directly related to *C* via inclusion and equivalence relations.

Formally, given an ontology concept *C*, the vicinity V_C of this concept includes a set of tuples *(C',S')*, where *C'* is the lexicalization of a concept directly related to *C* and *S'* is the WordNet sense that has been associated with this concept, or "*null*" in case there is no associated sense.Therefore, given a concept *C* and its vicinity V_C, the semantic morphism f_s computes S_C, which is the highest-ranked WordNet sense associated to *C* i.e. $f_s(C,V_c)=S_C$ where $V_C=\{(C_i,S_i)|C_i$ is in the vicinity of $C,\}$ and $f_s(C_i,V_i)=S_i$ where V_i is the vicinity of C_i. S_C is assumed to express the intended interpretation of the concept specification.

Using the algorithm in Fig. 3 for the concepts in an ontology, each ontology concept is associated with a set of graded WordNet senses. For instance, the concept "*facility*" is associated with the five senses that WordNet assigns to the term "*facility*". These senses range from "*something created to provide a service*" to "*a room equipped with washing and toilet facilities*". The highest graded sense $S_{facility}$ expresses the intended interpretation of the concept "*facility*" in the context of the given ontology.

It must be emphasized that although LSI exploits structural information of ontologies and WordNet, it ends up with semantic associations between terms. The algorithm is based on assumptions that influence the associations produced [7].

Using the intended meanings of the formal concepts, HCONE-merge constructs an ontology $O^n=(S^n, A^n), n=1,2,$ where, S^n includes the lexicalizations of the senses associated to the concepts of the ontology $O_n=(S_n, A_n), n=1,2,$ and A^n contain the translated inclusion and equivalence relations between the corresponding concepts. The ontology O^n is considered to be part of the hidden intermediate ontology. The construction of the intermediate ontology (by mapping the concepts of both original ontologies to WordNet senses) together with the minimal set of translated axioms results in ontologies' merging [7].

Given two ontologies O_1 and O_2 to be merged, and due to the crucial role of uncovering the intended meaning of concepts to the HCONE-merge method, we aim at automating the mapping of O_1 and O_2 to WordNet senses.

Table 1. Comparison of the mapping methods for the HCONE-merge

	Fully-Automated	User-validated	Semi-Automated
Percentage of concepts validated by the user (in the best case)	0%	≤100%	>0%
"Correct" mappings produced (in the best case)	80%	90%	90%

The goal is to achieve high precision in mapping concepts to their intended meaning with minimum human intervention.

Based on the algorithm for computing the s-morphism, we have shaped methods to ontology mapping, where human inspection and validation has been reduced down to the number of algorithm runs needed to correct the concept pairs whose associations produce inconsistencies with respect to the WordNet inclusion relations [8].

Table 1 compares the proposed methods [8] according to the amount of the automation they achieve and the "correct" mappings produced. The fully automated method requires the minimum number of user actions, but at the same time it achieves the lowest percentage of correct mappings. This is an iterative method that in each iteration re-computes concept mappings given the WordNet senses associated to the concepts during the last iteration. This approach is "unstable", given that correct mappings computed during an iteration may result to non-correct mappings when re-computed in the next iteration and so on. Therefore, this method does not guarantee to converge to a set of concept mappings.

On the other hand, the user-based method achieves higher percentage of correct mappings, but the actions that are required by the user imply considerable effort, since the user has to validate the mapping of each ontology concept. It must be pointed that this case requires also a considerable number of additional algorithm runs, equal to the percentage of wrong mappings. The semi-automated method however, in addition to the high percentage of correct mappings can significantly reduce the number of concepts that need validation by the user. However, in the worst case, where each concept is involved in at least one inconsistency, validation of all concepts is required.

4 Approximating the intended interpretations iteratively

The semantic morphism can be considered as a similarity function between ontology concepts and WordNet senses. For the computation of this similarity, as already explained, the s-morphism takes into account the vicinity V_C of each ontology concept *C*. The vicinity includes the concepts directly related to *C,* together with their intended meaning,

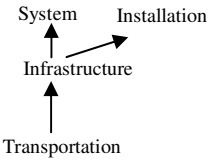


Fig. 4. Example ontology

For instance, to compute the intended interpretation of the concept "*Infrastructure*" of the ontology depicted in Fig. 4, the algorithm has to take into account the intended meanings of the concepts "*Installation*", "*System*" and "*Transportation*". However, to

compute the intended meaning of "*Installation*", the algorithm has to take into account the intended meaning of the concept "*Infrastructure*". As it is pointed in [10], this recursive dependency requires non-standard computation means. This problem has been approached by Bisson [11] and Euzenat [10] as an equation system where the similarity values are the solutions.

Given the ontology O_1 in Fig. 4, the following system of equations has to be solved:

1. $f_s(system,\{(infrastructure, S_{infrastructure})\})=S_{system}$
2. $f_s(installation,\{(infrastructure, S_{infrastructure})\})=S_{installation}$
3. $f_s(infrastructure,\{(system,S_{system}),(installation,S_{installation}),(transportation,S_{transportation})\}=S_{infrastructure}$
4. $f_s(transportation,\{(infrastructure, S_{infrastructure})\})=S_{transportation}$

As it has been proposed in [10], given the recursive nature of these computations, we can still find the intended meaning of each concept through an iterative process that finds the most nearest reachable fixed point of a vector function. The iteration produces a sequence of vectors of tuples $((C_1,S_1)\ldots (C_n,S_n))$, where each vector is an even more precise approximation of each concept's intended meaning.

Given the above formalization, the algorithm proposed in [10] works as follows: The initial approximation is based on the lexicalization of each concept (i.e. on 0-level contributors). The approximations at step $(n+1)$ are computed using the vicinities computed in step n.

Using a variation of the above algorithm, the intended meanings of concepts are computed iteratively as shown in Table 2:

Table 2. The computation of the approximation

Repeat the following process until there is no change in the intended meaning of any concept in the ontology. 1. For each ontology concept *C* do the following: 1.1. For each concept in the vicinity *V* of *C* In case there is no meaning associated to this concept compute the initial approximation based on its lexicalization 1.2. Repeat the following until there is no change in the approximation computed for the concept *C* 1.2.1 Compute the mapping of *C* using the approximations of concepts in *V* 1.2.2 Re-compute the approximations of concepts in *V* changing only the approximation of *C*

The computation of the approximation of each concept's meaning (i.e. the internal loop in 1.2. that computes the approximated meaning of a concept *C* based on the concepts in its vicinity) converges after two or three iterations. According to our experiments, independently of the size of the ontology, the algorithm finds a fixed point for the set of concepts in the ontology in the second iteration, improving the precision of the resulted mappings.

5 Results

We have run experiments, using the proposed algorithm, with the ontologies shown in Fig. 5, Fig. 6, and Fig. 7.

For instance, running the algorithm for the version of the Transportation ontology O_1 in Fig. 5, we have observed the following results for the concept "*car*": In the first approximation for this concept, given the initial approximations of all concepts in its vicinity, the algorithm computed the semantic morphism:

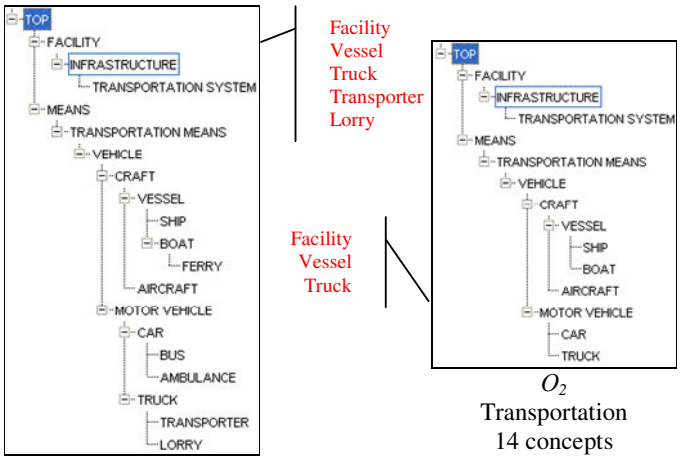
$$f_s(Car,\{(MotorVehicle, S_{MotorVehicle}), (Ambulance, S_{Ambulance}), (Bus, S_{bus})\}$$

The computation of this morphism returned the sense:

S_{car} = car, auto, automobile, machine, motorcar -- 4-wheeled motor vehicle,

In the second run for this concept, the algorithm found the same result, and therefore, this has been assumed to be concept's "*car*" intended meaning, given the meanings of the concepts in its vicinity. Traversing the ontology for a second time and re-computing the semantic morphism for the concept "*car*", given the new approximations of the concepts in its vicinity, the following results were returned for each iteration:

S'_{car} = cable car, car -- a conveyance for passengers or freight on a cable railway;
S'_{car} = car, auto, automobile, machine, motorcar -- 4-wheeled motor vehicle;



O_1
Transportation
19 concepts
73% precision

O_2
Transportation
14 concepts
78% precision

Fig. 5. Ontologies O_1, O_2

The precision for "uncovering" concepts' intended interpretation (in comparison to the meaning given by the engineer that devised these ontologies) is shown under each ontology snapshot. Concepts with incorrect mappings are shown within callouts drawings, attached to each snapshot.

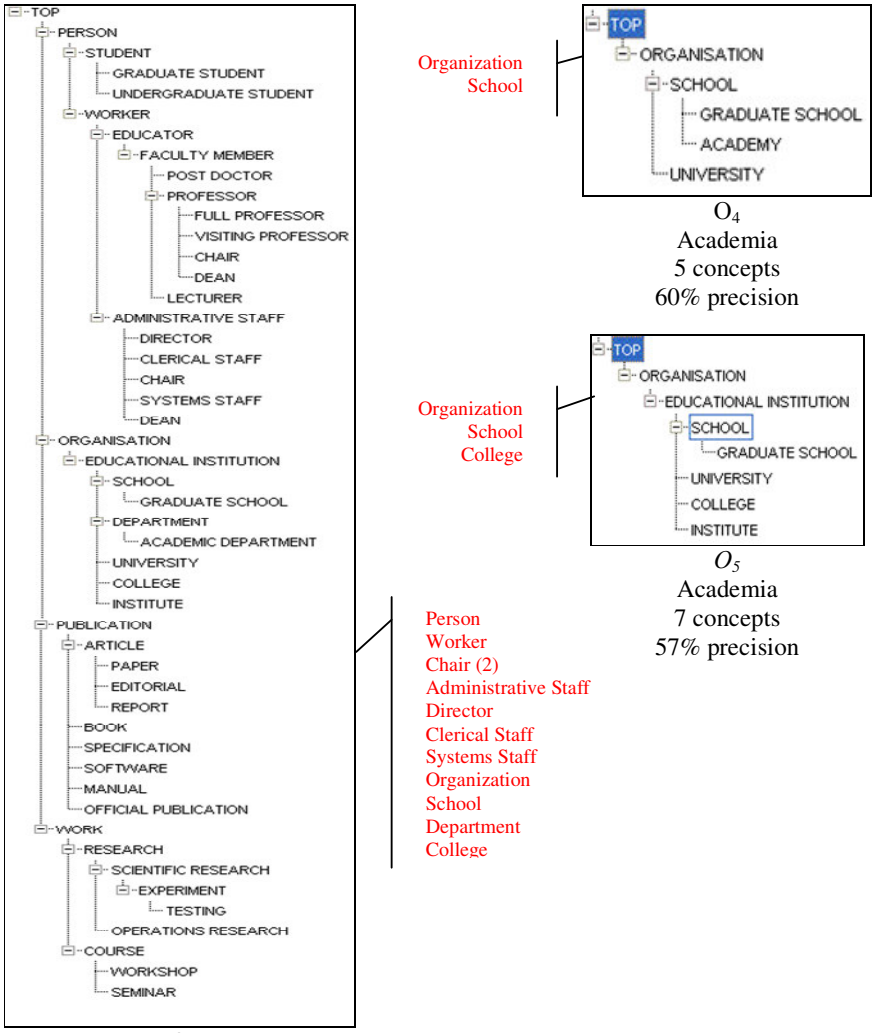
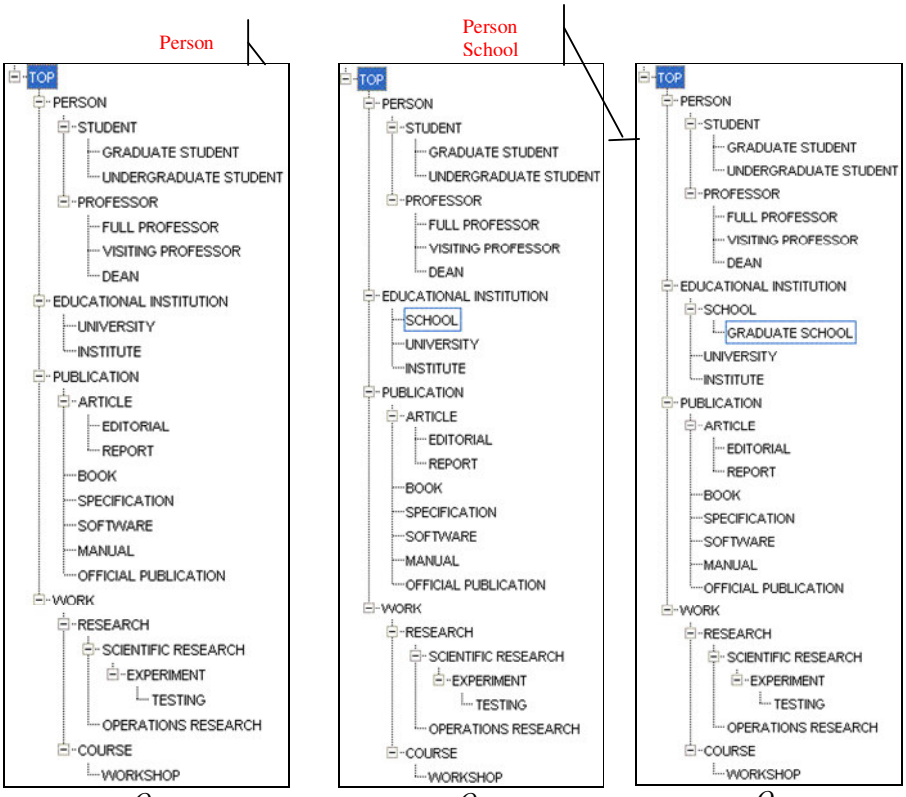


Fig. 6. Ontologies O_3, O_4, O_5



O_6
Academia
27 concepts
96% precision

O_7
Academia
28 concepts
92% precision

O_8
Academia
29 concepts
93% precision

Fig. 7. Ontologies O_6, O_7, O_8

The computation for this concept converged in the third run. So, even if the first sense S'_{car} was different (and it is not the intended one) from the one found in the first iteration for the ontology, the algorithm converges again after two iterations to the intended concept meaning.

For the same ontology, similar results are given for the concepts "*means*", "*infrastructure*", and "*craft*". The rest of the concepts do not change in the second iteration for the ontology. Therefore, the algorithm converges to a fixed point solution for the set of concepts in the second iteration for the ontology.

From the first experiments with ontologies O_1, O_2, and O_3, an average precision of 74% was concluded. However, we have been experimented with variations of these ontologies in order to investigate the behaviour of the algorithm in a controlled manner.

The ontologies O_4 and O_5 include a small set of concepts for which the iterative algorithm did not converge to their intended meaning. For the ontology O_4 these are the concepts "*organization*" and "*school*" and for the ontology O_5 the same concepts in addition to the concept "*college*". The rest 3 ontologies O_6, O_7 and O_8 are variations of the ontology O_3, and include a small percentage of concepts for which the algorithm did not compute their intended meaning (Fig. 7).

In more detail, the low precision achieved for the ontologies O_4 and O_5 is due to the failure of the algorithm to compute the correct mappings for the concepts "*organization*", "*school*" and "*college*". These concepts have not been mapped to their correct senses for O_3 as well. To explore the case that some concept mappings to WordNet cannot be "uncovered", we have experimented with different variations of the O_3 ontology. For instance, we have run experiments with variations of O_3 that include concepts from ontologies O_3, O_4, and O_5, which have been correctly mapped to WordNet, together with concepts whose computed intended meanings are not correct. Such a variation is the ontology O_6. This includes the concept "*person*" of O_3, whose mapping was not correct. The precision of mapping the ontology O_6 to WordNet is 96%, due to the wrong mapping of the concept "*person*".

The addition of the concept "*school*" – chosen from the set of concepts of O_3 with wrong mappings - to ontology O_6, results to the ontology O_7. The precision of the algorithm for this ontology is 92% due to the wrong mappings of the concepts "*person*" and "*school*".

Given that the sense $S_{Graduate\ school}$ computed for the concept "*graduate school*" in O_3 is the correct one, we may add this concept in ontology O_7 as a sub-concept of "*school*". This has happened in ontology O_8 where the vicinity of the concept "*school*" has been increased with the concept "*graduate school*". Although the algorithm computes the correct mapping for the concept "*graduate school*" in this new ontology, the algorithm computes a non-intended meaning for the concept "*school*".

The above experiments show that the performance of the algorithm can not be improved, even if the concepts in the vicinity of the ontology concepts with wrong mappings increase. The same happens even when the "distant" concepts (not included in the vicinity) change.

6 Conclusions

Towards automating the HCONE-merge method for merging ontologies, this paper proposes a method for aligning the original ontologies with a hidden intermediate ontology in a fully automated way. Actually, the alignment is done by mapping ontology concepts to WordNet senses. These senses are supposed to express the human oriented informal intended meanings of ontology concepts.

The algorithm proposed is based on previous efforts to approximate similarities between concepts in an iterative way and, as it has been shown, produces mappings that are quite precise. Furthermore, the algorithm converges fast; in two or three iterations requiring no extensive computational time.

The results provided from our case studies show that the algorithm behaves according to our intuitions and is stable: The computations it produces do not change in case the vicinity of a concept does not change radically and do not change even if the ontology, but not the vicinity of the concept, changes. Therefore, the computation for an ontology is not drastically affected by "distant" concepts. This result agrees with the requirement on dependencies between concepts' similarities within ontologies: The matching of a pair of concepts must depend on their local context and not to the entire ontology. However, concepts in the vicinity of a concept *C* must gather information from their own vicinity and further contribute such information to the computation of *C's* intended meaning.

Problems arise from the stability of the algorithm even for these concepts whose mapping is not correct. Experiments so far have not shown the exact reason for this to happen. However, future work concerns the combination of different information sources, except WordNet, for computing the intended meaning of such concepts.

References

1. Uschold M. and Gruninger M.: Creating Semantically Integrated Communities on the World Wide Web. Invited Talk, Semantic Web Workshop, WWW 2002 Conference, May, (2002)
2. Giunchiglia F. and Shvaiko P., Yatskevich M.: S–Match: An Algorithm and Implementation of Semantic Matching. The Semantic Web: Research and Applications, Lecture Notes in Computer Science, Vol. 3053, Springer-Verlag, (2004) 61-75
3. Kalfoglou Y. and Schorlemmer M.: Ontology mapping: the state of the art. The Knowledge Engineering Review 18(1):1-31 (2003)
4. Madhavan J., Bernstein P. A., and Rahm E.: Generic schema matching with Cupid. VLDB Journal (2001) 49-58
5. Doan A., Madhavan J., Domingos P., and Halvey A.: Learning to map between ontologies on the semantic web. In Proc. Of WWW-02, 11th InternationalWWW Conf., Hawaii (2002)
6. Noy N. and Musen M.: A. M.: PROMPT: Algorithm and tool for automated ontology merging and alignment. In Proceedings of 7th National Conference on AI, Austin (2000)
7. Kotis K. and Vouros G. A.. HCONE-Merge approach to ontology merging. The Semantic Web: Research and Applications, Lecture Notes in Computer Science, Vol. 3053, Springer-Verlag (2004) 137-151
8. Kotis K., Vouros G. A., Stergiou K.. Capturing Semantics towards Automatic Coordination of Domain Ontologies. To appear at the 11th International conference of Artificial Intelligence: Methodology, Systems, Architectures - Semantic Web Challenges - AIMSA 2004, Varna, (2004)
9. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman. Indexing by Latent Semantic Analysis. Journal of the American Society of Information Science (1990)
10. Jérôme Euzenat, Petko Valtchev, Similarity-based ontology alignment in OWL-Lite, Ramon López de Mantaras, Lorenza Saitta (eds), Proc. 16th european conference on artificial intelligence (ECAI), Valencia (ES), pp333-337, 2004
11. Gilles Bisson. Learning in FOLwith similarity measure. In Proc. 10th American AAAI conference, San-jose (CA US), 1992
12. Ghidini C., Giunchiglia F. A semantics for abstraction. In *Proceedings of the 16th European conference on Artificial Intelligence (ECAI-04)* Valencia, 22-27 August 2004