Exploring Web Semantic Knowledge and User Feedback to Improve Ontology Matching

Thiago Pachêco Andrade Pereira
Center for Informatics
Federal University of Pernambuco (UFPE)
Recife, Brazil
tpap@cin.ufpe.br

Carlos Eduardo Pires
Computer Science Department
Federal University of Campina Grande (UFCG)
Campina Grande, Brazil
cesp@dsc.ufcg.edu.br

Ana Carolina Salgado
Center for Informatics
Federal University of Pernambuco (UFPE)
Recife, Brazil
acs@cin.ufpe.br

***Abstract*—The first step to integrate multiple data sources is to decrease the heterogeneity between their schemas. This task can be facilitated if data source schemas are represented as ontologies. In this case, ontology matching enables the identification of correspondences between elements of data source schemas. We propose an ontology matching approach to improve the accuracy of correspondences using an additional source of knowledge. We take advantage of the knowledge available on the Semantic Web obtaining an ontology to be used as background knowledge. Semantic rules are executed in order to discover new correspondences between ontology elements. Our approach also offers to the users the possibility of rejecting invalid correspondences. These rejections are stored and used on further executions of the ontology matching to remove invalid correspondences.**

Keywords: ontology matching; semantic web; user feedback; background knowledge

I. INTRODUCTION

The first step to integrate multiple data sources is to decrease the heterogeneity of their schemas [1]. This task can be facilitated using ontologies to represent schemas and executing an ontology matching process over them. A schema matching process produces a set of correspondences between elements of the involved schemas. Many techniques were developed trying to improve the accuracy of correspondences exploring linguistic, structural and semantic approaches.

Some of the current challenges in ontology matching are described by Euzenat in [2]. Among these challenges, we can highlight the use of the Internet to discover additional knowledge. One possible approach is to use the ontologies available on the Semantic Web to discover new relationships between data source concepts.

In general, the knowledge contained on the compared ontologies is not sufficient to enable the inference of all the possible correspondences between ontologies. To overcome this problem, we search for additional knowledge on ontologies available on the Internet.

The Semantic Web is becoming more and more a reality [12] as the number of available ontologies increases. The Internet is achieving the Semantic Web goal of becoming a web of knowledge. However, the task of finding all of this knowledge is not easy. To facilitate this task, ontology crawlers such as Watson [12] and Swoogle [13] can be used. They provide public ontologies on the Web and offer an interface that allows the search for ontologies using keywords. Swoogle indexes words inside ontologies to enable user searches. Watson uses reasoners to extract semantic data and improve its indexing quality. It parses the obtained ontologies and look for inconsistencies in the relations.

Another important challenge of ontology matching consists in taking advantage of user feedback to improve the accuracy of an ontology matching result without being too intrusive [2]. This task becomes more challenging if we decide to learn from these feedbacks and use them in future matching operations. A few works describe alternatives to involve the user in the ontology matching process but their solutions are quite intrusive [5,6].

This paper proposes an approach to improve the accuracy of the alignments generated by an existing ontology matching process called SemMatcher [7]. In other words, we extend an existing process to use ontologies available on the Semantic Web in order to find new correspondences between concepts and properties that are not discovered by the current process. In addition, we use user feedback to identify incorrect correspondences and ignore wrong correspondences in future matching operations.

The work is organized as follows. Section II describes the ontology matching process to be extended. Section III describes our approach to search for new knowledge on the Semantic Web. Section IV explains how we obtain user feedback and use it to remove incorrect correspondences. Finally, Section V explains the implementation issues and presents the results of our experiments.

II. CURRENT ONTOLOGY MATCHING PROCCESS

Our current ontology matching process [7] uses a semantic-based approach, which brings together a combination of linguistic, structural and semantic matching techniques as shown in Figure 1. The process takes as input the two ontologies to be matched (O_1 and O_2) as well as a domain ontology (DO) to serve as background knowledge. All ontologies are represented in the Web Ontology Language (OWL). A linguistic-structural matcher and a semantic matcher are executed in parallel generating the alignments A_{LS} and A_S, respectively. To generate the alignment A_{LS}, any existing linguistic-structural matcher can be used.

1529-4188/11 $26.00 © 2011 IEEE
DOI 10.1109/DEXA.2011.31

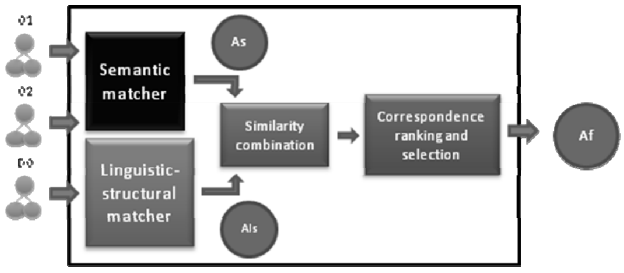

Figure 1. Ontology matching process overview before the improvements.

The semantic matcher [7] uses a semantic rule engine to infer semantic relationships using the domain ontology, as shown in Figure 2. There are seven possible types of relationships [8], each one associated to an arbitrary weight that represents the strength of the relationship:

a) IsEquivalentTo: If O_1:x points to a concept k in the DO and O_2:y points to the same concept k then O_1:x is equivalent to O_2:y. The similarity weight is 1.0.

b) IsSubConceptOf: If O_1:x points to a concept k in the DO, O_2:y points to another concept z in DO and DO:k is subconcept of DO:z then O_1:x is a subconcept of O_2:y. The similarity weight is 0.8. This rule is illustrated in Figure 2.

c) IsSuperConceptOf: If O_1:x points to a concept k in the DO, O_2:y points to another concept z in DO and DO:z is a subconcept of DO:k then O_1:x is a superconcept of O_2:y. The similarity weight is 0.8.

d) IsPartOf: If O_1:x points to a concept k in the DO, O_2:y points to another concept z in DO and DO:k is part of DO:z then O_1:x is part of O_2:y. The similarity weight is 0.3.

e) IsWholeOf: If O_1:x points to a concept k in the DO, O_2:y points to another concept z in DO and DO:z is part of DO:k then O_1:x is whole of O_2:y. The similarity weight is 0.3.

f) IsCloseTo: If O_1:x points to a concept k in the DO, O_2:y points to another concept z in the DO. DO:z and DO:k have a common ancestor DO:a, and their distance to DO:a is below a thresholdCommonAncestor then O_1:x is close to O_2:y. The similarity weight is 0.7.

g) IsDisjointWith: If O_1:x points to a concept k in the DO, O_2:y points to another concept z in the DO and DO:k is owl:disjointWith DO:z then O_1:x is disjoint with O_2:y. The similarity weight is 0.0.

The obtained similarity value of both matchers (linguistic-structural and semantic) are combined through a weighted average generating a combined alignment. Each matcher receives a particular weight according to its importance to the matching process. The correspondences of the combined alignment are ranked in descending order according to its associated similarity value. For each element on O_1, the most suitable correspondence is selected (i.e. the one with the highest similarity value). We denote this intermediate alignment A_{12}. Then, for each element on O_2, the most suitable correspondence is also selected, and we denote this alignment A_{21}. Afterwards, we have the most suitable correspondence for each concept of O_1 and O_2. The final alignment A_F is the union of A_{12} and A_{21}. These alignments are also used to calculate the global similarity measure between O_1 and O_2 using existing measures such as *dice* [9], *weighted* [10] and *overlap* [11]. The global similarity measure is a real number between 0.0 and 1.0.

The final alignment A_F and the global similarity measure are stored in a database. Each correspondence is stored as a 5-tuple (e, e', r, w, n), where e is an element of O_1; e' is an element of O_2; r is the semantic relationship between e and e'; w is the weight of the relationship and indicates how strong is the relation between the elements; and n is the level of confidence in this correspondence. The level of confidence is also a real number between 0.0 and 1.0. It will be explained in Section IV.

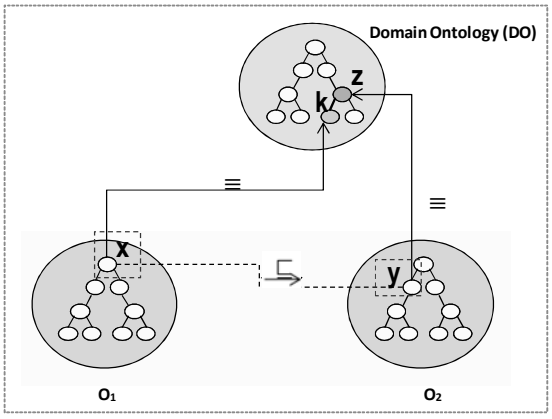

Figure 2. Example of how to find similar alignments.

III. USING THE SEMANTIC WEB AS BACKGROUND KNOWLEDGE

As a first step to enhance the matching process, this work proposes the use of a thesaurus to find synonyms and compare the concepts not only using their own labels to match, but also analyzing their synonyms. To this end, we use WordNet [14], a lexical database, containing English nouns, verbs, adjectives, and adverbs, organized into sets of synonyms, each one representing a lexicalized concept.

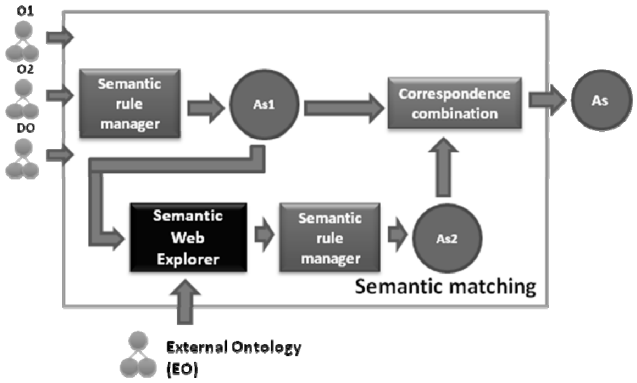

Figure 3. Semantic matching process using external knowledge.

Another improvement is to find new correspondences for the concepts that are not in the generated alignment A_{S1}. The label of the concepts that are not in A_{S1} will be used as keywords to find a ontology in Watson or Swoogle which may contain this missing knowledge. We use the concepts that do not have a correspondence because our goal is to find new knowledge about these concepts. We assume that the concepts already matched are solved. Watson and Swoogle return an ordered list of ontologies. The best ranked ontology is used as input for a second execution of the semantic rule manager (Figure 3) generating the alignment A_{S2}.

The new correspondences on the alignment A_{S2} are used to enrich the current alignment A_{S1}, generating the alignment A_S. As explained in Section II, a level of confidence is associated to each correspondence. In our work, we assign the value 1.0 to correspondences identified using the domain ontology (DO) provided by the user, and 0.8 to correspondences determined using ontologies downloaded from the Internet, but these values can be configured by the user. A higher level of confidence is assigned to the correspondences identified using the DO because we assume that an ontology provided by the user is more reliable than an ontology obtained in the Internet.

IV. Improving Results with User Feedback

When dealing with external knowledge, some incorrect correspondences can be found. The ontology obtained from the Internet or even the domain ontology may contain some incorrect knowledge. To overcome this problem, we allow the user to reject invalid correspondences when the matching process is finished. This information is stored in a database to be used in future matching operations to remove wrong correspondences automatically. If a user rejects a correspondence, we assign the value 0.0 to the level of confidence since, for that execution, the correspondence is invalid and there is no guarantee on the confidence of the correspondence.

In a further matching operation, to calculate the level of confidence, previous alignments that are similar to the current one are searched in the database. In other words, only the alignments involving ontologies that are similar to the current matched ontologies are verified. To compare the ontologies, we check in the database if the global similarity measure between those ontologies have been previously calculated and determine if the measure is higher than a certain user defined threshold which we call t_{sim}.

Figure 4 shows how we identify similar alignments. Considering the current matching operation between the ontologies O_3 and O_4, we compare this alignment against an older alignment A_1 (between O_1 and O_2) by checking the global similarity between the compared ontologies O_1 and O_2. We obtain these similarities consulting older alignments A_2 (between O_1 and O_3), and A_3 (between O_2 and O_4). If these similarities are higher than the user defined threshold t_{sim}, we use the alignment O_1-O_2 to analyze its correspondences.

Once previous similar alignments are identified in the database, for each correspondence in the current alignment, we calculate the new confidence level. To this end, we use a weighted average between the current level of confidence and the average of the previous levels of confidence. If the new level of confidence is below a user defined threshold t_{conf}, we discard the correspondence in the current alignment.

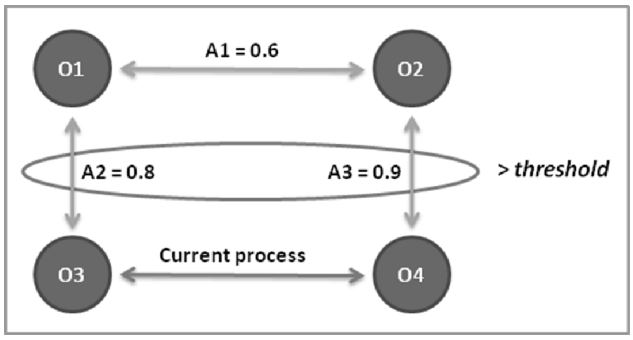

Figure 4. Example of how to find similar alignments.

V. Experiments

In order to show how the proposed approaches influence the generation of ontology alignments, we describe how our solution is integrated to the previous version of SemMatcher and compare the results obtained in both versions. We did not use the OAEI [17] benchmark because we deal with other kinds of relationships (e.g. *closeness* and *part-of*) which are not included in its reference alignments. Then, it would be unfair to compare the OAEI alignments with our results.

A. Implementation

The current implementation was made over the previous version of SemMatcher. It uses Jena [15] as a reasoner to execute the semantic rules and manipulate ontologies. The linguistic-structural matchers used are AlignmentAPI [18] and H-match [19]. However, it is possible to provide the correspondences through a text file using the alignment format defined by OAEI. The APIs offered by Swoogle and Watson were used to find external ontologies.

The generated alignments and the user's rejected correspondences are stored in a MySQL database. When checking synonyms on the WordNet, we used the MIT Java WordNet Interface (JWI) [16], a library which allows the access of the taxonomy and the synonyms. SemMatcher is used inside the SPEED system [8], a Peer Data Management System (PDMS) in which each peer is an autonomous data source that makes available a local schema represented as an ontology.

B. Results

We created two scenarios in order to execute our experiments. The first one makes use of external ontologies whilst the second uses user feedback to improve the results. Both scenarios use WordNet to help the process of finding equivalent elements between the input ontologies and the domain ontology.

1) Scenario 1: Using ontologies available on the Web

To evaluate the accuracy of the alignments, we consider the classical measures precision and recall. These measures are suggested by OAEI to evaluate ontology matching tools. To calculate them, we use a reference alignment, which contains

the correct correspondences. Additionally, we analyze the elapsed time and the number of correspondences generated.

Ontologies belonging to the education domain were used. When searching for external ontologies, we noticed that large ontologies can be found (for instance, 20MB), and the download of these ontologies brought some performance problems. Thus, in our tests, we considered the possibility of searching for external ontologies with an unlimited size and with a limited size of 5MB. A comparison between the two versions of SemMatcher is shown in Table I.

TABLE I. COMPARISON ACCURACY BETWEEN THE VERSIONS.

	Precision	Recall	Time	#Corresp.
Old version	0.85	0.94	10s	158
New version with unlimited ontology size	0.90	0.94	2m35s	173
New version with limited ontology size	0.86	0.96	41s	162

A large ontology causes a performance problem not only to download it but also to load the ontology into the SemMatcher. This is confirmed by the wasted time of the second execution, in which we did not limit the size of the external ontology. The time to download the ontology is another problem as it depends on the network bandwidth.

Considering the execution with an unlimited ontology size, the value of precision increased, showing that the new correspondences found are correct. Analyzing the number of correspondences in the final alignment we can notice that the number of correspondences increased almost 10% (see Table I).

The test using ontologies with limited size has also found four new correspondences, not as much as the one with an unlimited size. On the other hand, the performance has improved and the process was executed in less than half of the time.

2) Scenario 2: Using user feedback to eliminate invalid correspondences

To show how user feedback affects the level of confidence, we simulate consecutive executions of SemMatcher (in a real scenario, it would happen with different users). In the simulation, we analyze a single correspondence which we are sure that it will appear on the ontologies to be matched.

The correspondence is ⟨*Accepted_Paper*, *isCloseTo*, *Rejected_Paper*⟩. We know that this correspondence is invalid because *Accepted_Paper* is disjoint with *Rejected_Paper*. Table II shows how the level of confidence on this correspondence is updated after successive executions of the process until it is eliminated.

TABLE II. LEVEL OF CONFIDENCE AFTER MULTIPLE FEEDBACKS.

Execution	Average from the previous alignments	Current level of confidence	New level of confidence	User feedback	Confidence after feedback
1	-	1	1	Not availed	1
2	1	1	1 x 0.4 + 1 x 0.6 = 1	Rejected	0
3	0.5	1	0.5 x 0.4 + 1 x 0.6 = 0.8	Rejected	0
4	0.33	1	0.33 x 0.4 + 1 x 0.6 = 0.73	Rejected	0
5	0.25	1	0.25 x 0.4 + 1 x 0.6 = 0.7	Rejected	0
6	0.2	1	0.2 x 0.4 + 1 x 0.6 = 0.68	(Correspondence rejected because of the *threshold*)	0

When a user rejects a correspondence, its level of confidence is updated to 0.0. For example, in the first execution, the initial level of confidence is 1.0, since the correspondence was identified using an ontology provided by the user. There are no old similar alignments in the database and thus no extra processing is done. Since the user did not reject the correspondence, the confidence value remains 1.0.

In the second execution, using ontologies similar to the first ones, the same correspondence is found and the current level of confidence is 1.0 since it was found in an ontology provided by the user. Then, the new level of confidence is calculated with a weighted average between the average level of the older similar alignments (in this case, the alignment of the first execution) and the current level of confidence. To calculate the weighted average, we considered the weight 0.4 to the average of older levels of confidence, and 0.6 to the current level of confidence. In this execution, the value is still 1.0. However, after the alignment is generated the user rejects the correspondence in a new feedback, and for this execution, the correspondence has its level of confidence updated to 0.0.

On the third execution, the current level of confidence is 1.0 because it is found using an ontology provided by the user. When calculating the new level of confidence we find that the average from older alignments is 0.5 (the average between executions 1 and 2), the weighted average is calculated with the current level and the new value is 0.8 and so on. When in some point, the level of confidence is lower than the user defined threshold t_{valid} (in this case we assume the user used 0.7), this correspondence is automatically eliminated from the final alignment. After multiple rejections of a correspondence, its level of confidence decreases until it becomes lower than t_{valid}. In this case, the correspondence is considered invalid and is eliminated.

VI. RELATED WORK

Some matchers such as SCARLET [3] and GeRoMeSuite [4] explore the Semantic Web to find additional knowledge to improve the generation of correspondences. WordNet is used in different ways by the matchers. S-Match [20] and C-SAW [21] use WordNet as background knowledge. Po and Bergamaschi [22] extend SCARLET to compare not only the exact word to find concepts in the ontologies found in the Internet, but also its synonyms. Among these matchers, only S-Match suggests the use of user feedback in future versions.

SemMatcher combines some of these approaches to find new correspondences using the Semantic Web. It also provides to users the possibility of informing wrong correspondences. This information is stored in a database and used to identify wrong correspondences in future executions.

VII. CONCLUSION

Finding alternative external sources of knowledge is useful to discover additional correspondences that would not be inferred using only local knowledge. Sites such as Swoogle and Watson provide a good interface to search for ontologies that can be used to improve an ontology matching process.

Our experiments showed that new correspondences were found when looking for ontologies with an unlimited size. These correspondences are accurate since they were verified using appropriate metrics (precision and recall). Another useful source of knowledge is the user. Commonly, the user is the only source that knows the correct correspondences. When informing invalid correspondences, we can use this knowledge to make the system learn about that correspondence and eliminate wrong correspondences in the future.

As a future work, we intend to search for external ontologies on the Internet using not only the concepts that do not belong to any correspondence, but also concepts that have a correspondence. The intention is to find ontologies with knowledge that we already discovered, to confirm it. Concerning the user feedback, we can improve the process by removing not only the invalid correspondences, but some related ones. For instance, if the following correspondences are true: $c_1 \subseteq c_2$ and $c_2 \subseteq c_3$, and we use them to infer $c_1 \subseteq c_3$. After the feedback of the user, if $c_1 \subseteq c_2$ is invalidated, we also remove $c_1 \subseteq c_3$ because it was inferred using a wrong correspondence. Finally, a specific experiment using the synonyms must be done in order to better evaluate this feature.

REFERENCES

[1] J. Euzenat, P.Shvaiko, “Ontology matching”. Springer-Verlag, Heidelberg, 2007.

[2] P. Shvaiko, J. Euzenat, "Ten challenges for ontology matching". In Proc. of the 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE), pp. 1164–1182, Monterrey, México, 2008.

[3] M. Sabou, M. d’Aquin, E. Motta, "Exploring the semantic web as background knowledge for ontology matching". Journal of Data Semantics XI, pp. 156-188, 2008.

[4] C. Quix, M. Pascan, P. Roy, D. Kensche, "Semantic Matching of Ontologies". In Proc. of the International Workshop on Ontology Matching, Shanghai, China, pp 19-20, 2010.

[5] H. Paulheim, M. Rebstock, Fengel. "Context-sensitive referencing for ontology mapping disambiguation". In Proc. of the Workshop on Contexts and Ontologies Representation and Reasoning, Roskilde, Denmark, pp. 47-56, 2007.

[6] S. Duan, A. Fokoue, K. Srinivas, "One Size Does Not Fit All: Customizing Ontology Alignment using User Feedback". In Proc. of ISWC, Shanghai, China, pp. 177-192, 2010.

[7] C. E. Pires, D. Souza, T. Pachêco, and A. C. Salgado, "A Semantic-Based Ontology Matching Process for PDMS". In Proc. of Globe’09, pp. 124-135, Linz, Austria, 2009.

[8] Souza, D. 2009. "Using Semantics to Enhance Query Reformulation in Dynamic Distributed Enviornments". Ph.D. Thesis, Federal University of Pernambuco (UFPE/CIn). Recife, PE, Brazil.

[9] D. Aumueller, H. Do, S. Massmann, E. Rahm, "Schema and ontology matching with COMA++". In Proc. of SIGMOD’05, pp. 14-16, Baltimore, Maryland, USA, 2005.

[10] S. Castano, V. Antonellis, M. G. Fugini, B. Pernici, "Conceptual Schema Analysis: Techniques and Applications". In ACM Transactions on Database Systems, Vol. 23, No. 3, pp. 286-333, 1998.

[11] C. J. Rijsbergen, "Information Retrieval". 2nd Ed. Stoneham, MA, Butterworths, 1979.

[12] T. W. Finin, L. Ding, R. Pan, A. Joshi, P. Kolari, A. Java, Y. Peng, "Swoogle: Searching for knowledge on the semantic web". In Proc. of the National Conference on Artificial Intelligence, Pittsburgh, Pennsylvania, pp. 1682-1683, 2005.

[13] M. d’Aquin, C. Baldassarre, L. Gridinoc, M. Sabou, S. Angeletou, E. Motta, "WATSON: Supporting next generation of semantic web applications". In Proc. of the WWW/Internet Conference, Vila Real, Portugal, pp. 20-28, 2007.

[14] C. Fellbaum, "WordNet: An Electronic Lexical Database". MIT Press, Cambridge, EUA, 1998.

[15] J. J. Carroll, I. Dickinson, , "Jena: Implementing the Semantic Web Recommendations". Technical Report HPL-2003-146, Hewlett Packard Laboratories, Bristol, 2003.

[16] “MIT Java Wordnet Interface (JWI)”. Available at http://projects.csail.mit.edu/jwi/. Last access: 18/03/2010.

[17] J. Euzenat, M. Ehrig, R. G. Castro, "Towards a methodology for evaluating alignment and matching algorithms". Technical Report, Ontology Alignment Evaluation Initiative (OAEI), 2005.

[18] J. Euzenat, "An API for ontology alignment". In Proc. of the International Semantic Web Conference, pp. 698-712, Hiroshima, Japan, 2004.

[19] S. Castano, A. Ferrara, S. Montanelli, "H-match: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems Peer-based Systems". In Proc. of the 1st VLDB Int. Workshop on Semantic Web and Databases, Berlin, Germany, 231--250, 2003.

[20] F. Giunchiglia, P. Shvaiko, M. Yatskevich. "Semantic matching with S-Match", Springer, 2009.

[21] B. Tierney, M. Jackson, "C-SAW-contextual semantic alignment of ontologies: using negative semantic reinforcement", In Proceedings of the 2008 ACM symposium on Applied computing, pp. 2346-2347, 2008.

[22] L. Po, S. Bergamaschi, "Automatic lexical annotation applied to the SCARLET ontology matcher", In Proceedings of ACIIDS, pp. 144-153, 2010.