LOM: A Lexicon-based Ontology Mapping Tool

John Li
Teknowledge Corporation
1800 Embarcadero Road
Palo Alto, CA 94303

ABSTRACT

Ontology mapping is important to knowledge sharing and semantic integration but hard to completely automate. LOM is a semi-automatic lexicon-based ontology-mapping tool that supports a human mapping engineer with a first-cut comparison of ontological terms between the ontologies to be mapped, based on their lexical similarity. This paper will explain the algorithms used, the tests performed, and the applications developed using the results of this approach. It will also discuss the limitations of this approach as well as the future research and development issues in this field.

KEYWORDS: *ontology, mapping, lexicon, semantic web, semantic integration, alignment, interoperability*

1. INTRODUCTION

Ontology mapping is an important step to achieving knowledge sharing and semantic integration in an environment in which knowledge and information have been represented with different underlying ontologies. As more applications exploit semantic interoperability by employing an increasing number of ontologies developed by diverse communities, the demand for rapid ontology mapping is arising. Many efforts have been spent on machine-assisted ontology mapping [1]. However, this task is by nature very difficult to automate because heterogeneous ontologies may reflect fundamentally or subtly different perceptions of the domain by the creators of these ontologies. The evidence for the difficulty in producing a fully automated method for ontology mapping can be traced back to an early survey on automated database schemata alignment and to a recent one on the state of the art in ontology mapping [1, 2].

We view ontology mapping as a learning process, by human or machine, to find a morphism between the concepts of the given ontologies. Given two ontologies, A and B, a mapping from A to B is a set of pairs *(a, b)* where *a* is a concept expressed in A and *b* is its translation in B. Note that *a* and *b* can be represented in terms or expressions. Obviously the mapping is partial and not necessarily one-to-one depending on the ontologies under consideration. A good mapping tool should find the maximal number of potential mapping pairs. Naturally, if there is no overlapping of concepts in the two ontologies, there is no mapping that can be found between them.

As ontologies are logical theories that contain vocabularies and axioms for concepts, the first step in ontology mapping is to find the morphism between their vocabularies. LOM was just designed for that purpose. It is a prototype lexicon-based ontology-mapping tool developed at Teknowledge, under the Agent Semantic Communication Services (ASCS) project [3] for DARPA Agent Markup Language (DAML) Program [4]. LOM supports a human mapping engineer with a first-cut comparison of ontological terms between the ontologies to be mapped, based on their lexical similarity. We call LOM a semi-automatic method because it requires human validation at the end of the process. The output of LOM, which is a list of matched pairs of terms with scores ranking their similarity, will be reviewed by the human for the final decision. The finally approved matched vocabulary will serve as the basis for the axiom translation.

The development of LOM was based on the following two observations: (1) Human intervention in ontology mapping cannot be totally avoided but human labor can be reduced by mechanic comparisons done by intelligent software, and (2) The lexicon-based mapping is feasible because most ontologies bear lexical similarity in their vocabularies describing the same concepts when the natural languages underlying the vocabularies are the same (such as English). This linguistic connection exists naturally since most ontologies are developed by humans and are required to be understood by both humans and agents. That provides a good opportunity for our software to explore the common language base of the heterogeneous ontologies and to use syntax and semantics to identify the similarity between the terms. Like most mapping tools, LOM does not guarantee accuracy nor correctness in its suggested mappings. It saves human labor by changing their job from tedious and time-consuming search and matching tasks to much easier ones of approval and validation.

This paper is a work-in-progress report since LOM is still under development. In the next section we will present the algorithms used in LOM (Section 2). Section 3 describes the results of some tests as well as some semantic web applications using the mappings developed

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
AUG 2004		**00-00-2004 to 00-00-2004**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
LOM: A Lexicon-based Ontology Mapping Tool	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Teknowledge Corporation,1800 Embarcadero Road,Palo Alto,CA,94303**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Proceedings of the 2004 Performance Metrics for Intelligent Systems Workshop (PerMIS -04), Gaithersburg, MD on August 24-26 2004

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**5**	

by LOM, followed by a discussion on the future development of LOM and possible improvements. Section 4 briefly reviews some related work. Section 5 contains a summary.

2. ALGORITHMS

LOM uses four methods to match the vocabularies from any two ontologies. They are (1) whole term matching; (2) word constituent matching; (3) synset matching; and (4) type matching. We will explain each method in detail below. As the first step, vocabularies should be separated into lists of classes, predicates and instances, and then compared class vs. class, predicate vs. predicate, etc. However, sometimes it is desirable to compare whole vocabularies without such classification since some authors may represent similar concepts with different types of terms.

LOM takes two lists of terms from ontologies A and B and produces a list of matched pairs. Each pair contains two terms: one from the source, A, and the other from the target, B. Each term can be multi-word, such as "BiologicalParent" or "office-phone-number", etc. The matched pairs are then found through the following procedures:

(1) Whole term matching: This is the first as well as the simplest procedure to be executed. The terms in both ontologies are converted to lowercase and then compared for an exact name string match. The matched pairs are given a score of 1. Otherwise, the score is zero.

(2) Word constituent matching: This is the second procedure to be executed. Each term is broken into words wherever there is a capital letter, a hyphen or an underscore. Stop words such as "a", "the", "of", "in", etc. are dropped from multi-word terms. Remaining words for each term are morphologically processed and compared in exact string match to words of each term from the target ontology. Every matched pair has a score from 0 to 1, inclusive, representing the ratio of the number of the words matched with regard to the total number of word constituents. Then, for each term, among all its matched pairs, only the *best-fit* pairs (the highest scorers) are recorded and presented to the user. Using this procedure, unobvious matching term pairs such as "written-by" and "wrote", "meeting-place" and "place-of-meeting" can be found.

(3) Synset matching: This is the third procedure to be executed. It explores the semantic meanings of the word constituents by using the WordNet [5] synsets to help identify synonyms in matching. A synset is a WordNet term for a sense or a meaning by a group of synonyms. This procedure is similar to the method in (2) in decomposing multi-word terms into their word constituents except that it does not perform direct matching between the words. For each word in each term in each ontology, if it is in WordNet, then it must belong to one of the synsets and have at least one WordNet synset index number. The procedure associates the WordNet synset index numbers of the constituent words with the term. The two terms which have the largest number of common synsets are recorded and presented to the user. Their score is calculated and recorded in the same method as that in (2). Using this procedure, the terms "auto-care" and "car-maintenance", for example, can be matched.

(4) Type matching: This is the last procedure to be called by LOM, and it explores the ontological category of each word constituent for matching. It uses the mappings from WordNet synsets to the formal ontologies SUMO (the Suggested Upper Merged Ontology) [6, 7] and MILO (the Mid-level Ontology) [8]. SUMO and MILO together contain about six thousand ontological terms at the upper and middle level. The most popular WordNet synsets have been mapped into this set of terms [9]. LOM takes the source terms that are unmatched in the above-mentioned three procedures, collects the set of SUMO/MILO terms that their synsets map to, and then compares the SUMO term sets to their counterpart for each term in the target ontology. If there is a match, the matched terms are recorded and given a score based on the method of calculation in (2) and (3). The matched terms with the highest score for each term are recorded. Using this procedure, terms that cannot be matched by previous methods, either string comparison or sense comparison, will be matched if they represent classes or properties of the same type. For example, the terms "tank" and "armed-personal-carrier" can be matched since they are both military vehicles.

There are several caveats about the methods we mentioned above. First, the morphological processes used in procedure (2) are standard for the English language and we will not describe them here. However, if other languages are used, the morphological processes need to be replaced with rules for the other languages. Second, to do an ontology mapping from A to B, each term in source A is tested against *every* term in target B. Thus the algorithm runs in $O(nm)$ time where n and m are the length of the two input term lists respectively. During the execution, the list in B does not decrease although that in A may, as the matched ones in the source may leave the game. Third, one may think the most efficient way to execute these four procedures is to follow the sequence and let each procedure process the leftover of the previous procedure. To determine what constitute the leftover, the user needs to determine the thresholds for all methods except (1), which has only two scores: 0 and 1. If the score of a matching pair is below that threshold, the source term in the pair will be left to the next procedure to continue the process. Finally, after all methods are applied, the leftover in the source list are

unmatched. Another way of executing these methods is to filter out the matched pairs after the first procedure is executed but leave those from the second or the third procedures in the game and let them do alternative matching. To help it, LOM identifies in its output the method it uses to reach the matching together with the score of the matching. One advantage of the second way of execution is that there is no need for the artificial thresholds. Either way, each procedure does not need to repeat the process done by the previous procedure, such as breaking-down the multi-word terms, morphologically processing words, and finding synsets, etc. The second way of execution creates more opportunities for the mapping but requires more time when the ontologies are big. Fourth, one may easily find that the precision of the matching differs from procedure to procedure. Obviously the mapping through type matching can be very inaccurate since there are a limited number of ontological categories at the upper and middle level. This method is used as the last resort.

Here we have presented an algorithm for LOM and explained some of its features. The whole software is implemented in Prolog. In the next section we will report some of the tests LOM underwent and some applications it had contributed to. We will talk more about the issues and possible improvements to LOM after that.

3. TESTS AND APPLICATIONS

3.1 *Tests*

LOM has been tested extensively in-house to evaluate its functionality and performance. As this paper is written, it is participating in a competition at I^3CON (Information Interpretation and Integration Conference) [10]. In its early development stage we had run an experiment with the test data created by the SENSUS development team at the Information Science Institute (ISI) of the University of Southern California [11]. The data consists of 102 pairs of matched terms between SENSUS and CYC. LOM took the terms from both the SENSUS ontology and the CYC ontology and generated a set of mappings that were compared to the manual mappings that ISI and Cycorp created by hand. Then, metrics like precision and recall as used in the information retrieval were computed. According to our calculation, precision was 54/76 (71%) and recall was 54/94 (57%) for this experiment. Note that in this experiment we were using an early version of SUMO and an incomplete mapping from the WordNet synsets to the SUMO, so the procedures (3) and (4) did not help much in the mapping. Following that we did many test runs with the ontological terms developed by the DAML ontology community. The metrics generally improved but still varied depending on the contents and the representations of the ontologies to be mapped. On the performance measure, the time to run inputs of about 100 terms per ontology is in seconds on a 500MHz laptop. The same machine with an increased RAM size (512MB) and an increased stack size can run inputs of over one thousand terms per ontology. The ability to perform a first-cut mapping on big ontologies has been the target of our performance improvement efforts because that ability is exactly the goal of the LOM development.

3.2 *Applications*

LOM is an important component in the ASCS [4] tool set. ASCS was intended to provide semantic search and translation functions to semantic web applications. Teknowledge's DAML/OWL [12] Semantic Search Service crawls web pages, gathers semantically marked contents into a repository, and provides a search engine that allows people to query the repository and get data as the answers to their queries. Its most recent version even allows people to publish their own data into the repository via URL registration, and to register their queries and get automatic notification when the conditions for the queries are met. Obviously, with such extensive and diverse authorship, the number of ontologies underlying the data is increasing steadily. Envisioning the massive growth of diverse ontologies, the OWL designers created a set of OWL terms such as "equivalentClass", "equivalentProperty" and "sameAs" to help the authors of the ontologies to align their creation with others. Our semantic search engine not only can use these relations to seek equivalent data but also can reason with other ontological relating predicates such as "subClassOf", "subPropertyOf" and "inverseOf" to perform semantic search.

Despite all these relating predicates and the superb search capability of our search engine, the semantic search remains a problem if the authors did not actually produce the equivalence instances using these predicates. Without these instances, the data would still be isolated islands. To a search engine developer, that means a query based on one ontology will not be able to match data across the ontology boundaries although they are semantically answerable. It can be an even severer problem for the semantic-search query language designers if they have to choose a certain ontology as the base for the query language because whatever ontology the query language is based on, the answers will stop within that ontology, if these instances do not exist. We took this opportunity to test the usability of LOM. With the help of LOM, we quickly located the matching pairs from a group of ontologies and generated a big set of equivalence instances over certain domains, such as bibliography and terrorisms. In the bibliography domain,

for example, we mapped six ontologies to SUMO and generated about 300 instances of equivalent classes and properties in a very short time period. As expected, these instances greatly expanded the search range of our search engine and enabled it to answer queries with data marked in different ontologies from diverse sources. In addition, they enabled us to reduce our query interface to a much simpler one. The users do not need to remember nor specify the multiple terms and the multiple ontologies they had to use when they formed the query because there is only one ontology underlying the query language. With our recently developed Restricted English Query Interface [13], the user only needs to enter a conjunct English query (using *What, Who, When, Where* and other regular English words) and the interface will translate it into a logic form based on the SUMO and execute it. Since our repository is populated with the equivalence instances relating terms from other ontologies to those in the SUMO, our search engine will be able to gather data from multiple diverse sources using these relations.

3.3 *Discussion*

Although the experiments and applications showed that LOM made contribution to the ontology-mapping tasks, we realize that there are many places where LOM can be improved. To strengthen the word constituent matching method, LOM needs to recognize proper names, shorthand and abbreviations correctly. For example, it may need to use some fuzzy syntactic analysis method to learn that "SemWeb" is the shorthand for "SemanticWeb". As more low-level domain ontological terms are being developed and deployed, the mappings from WordNet synsets to SUMO will be updated to achieve higher accuracy. Both the synset matching method and the type matching method can benefit from the enhanced accuracy and find closer sense or type matching between the terms. As a lexicon-based ontology mapping method, LOM has its limitation in handling ontologies built with abstract symbols or codes, such as those used in chemistry, mathematics, or medicine. We plan to implement a structural mapping method that may alleviate the weaknesses of the lexicon-based approach by recognizing structural similarity between the ontologies.

4. RELATED WORK

Information integration has been a research topic for the database and KR communities for many years. With the emergence of the Internet and the advent of DAML/OWL language, semantic interoperability issues and solutions are gaining a greater audience. Among the vast number of publications related to the ontology alignment, we recently found the proposed conceptual alignment process in [14] had suggested the usage of syntactic and lexical analyses for similarity measuring, similar to what we have developed for LOM, although the development of LOM started in 2001, one year earlier than the proposal was published. Besides the difference between a proposal and an implementation, LOM has an additional method - type matching. Nevertheless, this paper provides some ideas about the integration of different methods that, as well as those in [15], might help us to explore the future development of LOM. Among the similarity learning algorithms, we found the similarity flooding algorithm [16] might be useful to the future development of LOM. Multi-strategy learning for ontology mapping was explored in [17].

5. SUMMARY

We have developed a lexicon-based ontology-mapping tool as one of many approaches in the ontology mapping research and development arena. This approach explores the lexical similarities between ontological vocabularies by using its four matching procedures: whole term matching, word constituent matching, synset matching and type matching. We have used it in some experiments and some semantic web applications in which it showed its strengths and weaknesses. As we view ontology mapping as a machine learning process, we will use this tool as the starting point to pursue multi-strategy learning of similarities between the ontologies that will take advantage of the strengths of various approaches. We expect that there will be some research and development issues ahead of us before all the desirable features can be integrated into this tool. There are vast and important applications (such as semantic integration, semantic web services) for ontology mapping in the real world. We are looking forward to continuing our research in this field and the practical deployment of our mapping tool to serve real-world users.

6. ACKNOWLEDGEMENT

The algorithms used in LOM were designed with help from my former colleagues, Ian Niles and Adam Pease. I also wanted to thank Professor Eduard Hovy for his generosity in sharing his SENSUS mapping data with us to support our early experiment. The development of LOM was partially supported by the DARPA Agent Markup Language (DAML) program.

7. REFERENCES

[1] Kalfoglou, Y., and Schorlemmer, M., "Ontology mapping: the state of the art," *The Knowledge Engineering Review*, Vol. 18:1, pp. 1-31, 2003.

[2] Sheth, A. and Larson, J., "Federated database systems for managing distributed, heterogeneous, and autonomous databases," *ACM computing Surveys* 22(3) pp. 183-230 1990.

[3] The DARPA Agent Markup Language (DAML) program, Internet Location: http://www.daml.org

[4] Agent Semantic Communication Services (ASCS), Internet Location: http://reliant.teknowledge.com/DAML/

[5] Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. "Introduction to WordNet: An On-line Lexical Database," *International Journal of Lexicography* 3(4) pp. 235-244 1990.

[6] Niles, I., and Pease, A., "Toward a Standard Upper Ontology," *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems* (FOIS-2001), Chris Welty and Barry Smith, (Eds.). 2001.

[7] Pease, A., Niles, I., Li, J., "The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications", in Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web. 2002.

[8] Niles, I., and Terry, A., "The MILO: A general-purpose, mid-level ontology," *Proceedings of the International Conference on Information and Knowledge Engineering* (IKE'04), Las Vegas, Nevada. 15-19. 2004.

[9] Niles, I. and Pease, A., "Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology," In Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE'03), Las Vegas, Nevada, June 23-26, 2003.

[10] Information Interpretation and Integration Conf. (I^3CON), August 24-26, 2004. Gaithersburg, MD., Internet Location: http://www.atl.external.lmco.com/projects/ontology/i3con.html

[11] Hovy, E.H., "Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses," In Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC). Granada, Spain. 1998.

[12] OWL Web Ontology Language, Internet Location: http://www.w3.org/TR/owl-features/

[13] The Restricted English Query Interface, Internet Location: http://ibis.teknowledge.com:8080/DAML/query.jsp

[14] Silva, N. and Rocha, J., "Merging Ontologies using a Bottom-up Lexical and Structural Approach," *Proceedings of The Seventh International Society for Knowledge Organization Conference* (7th ISKO), Granada, Spain, July 2002.

[15] Ehrig, M. and Sure, Y., "Ontology mapping – an integrated approach," Internet Location: http://www.aifb.uni-karlsruhe.de/WBS/meh/mapping/combinationTR.pdf

[16] Melnik, S., Garcia-Molina,H., Rahm, E., "Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching," *Proceedings of The 18th International Conference on Data Engineering* (ICDE'02), San Jose, California, February 26 - March 01, 2002.

[17] Doan, A., Madhavan, J., Domingos, P., Halevy, A., "Ontology Matching: A Machine Learning Approach," *Handbook on Ontologies in Information Systems*, S. Staab and R. Studer (eds.), Springer-Velag, 2004. pp. 397-416.