Extended Abstract: A Tool For Mapping Between Two Ontologies Using Explicit Information*

Sushama Prasad, Yun Peng, and Timothy Finin
Computer Science and Electrical Engineering Department
University of Maryland Baltimore County
email: {sprasa2, ypeng, finin}@csee.umbc.edu

1 Overview

Understanding the meaning of messages exchanged between software agents has long been realized as one of the key problems to realizing multi-agent systems. Forcing all agents to use a common vocabulary defined in a shared ontology is an oversimplified solution, especially when these agents are designed and deployed independently of each other. An alternative, and more realistic, solution would be to provide mapping services between different ontologies (Weisman, Roos and Vogt[5], Pinto[2]). In this paper, we present our work along this direction. This work combines the recently emerging semantic markup language DAML+OIL (for ontology specification), the information retrieval technique (for similarity information collection), and Bayesian reasoning (for similarity synthesis and final mapping selection), to provide ontology mapping between two classification hierarchies.

The two hierarchies we used as examples are ACM topic ontology and a small ITTALKS topic ontology which organizes classes of IT related talks in a way different from ACM classification. Both ontologies, as well as the output mappings, are marked up in DAML+OIL[1]. These two ontologies are extended by attaching to each concept/class a set of *exemplar*s, which are URLs pointing to the locations of text documents thought to belong to that class.

A *model* is built for each ontology, which primarily contains statistical information about the exemplar documents associated with each concept in that ontology, using the Rainbow text classifier[2]. Then, each concept of one ontology is mapped into one or more concept of the other ontology by comparing it's exemplars against the other ontology's model, again using Rainbow classifier. The raw similarity scores returned by the classifier are used by the mapper to produce a set of possible mappings between the two ontologies.

Based on the subsumption operation in description logics, two algorithms have been developed to synthesize the raw similarity scores toward the final mappings. One is based on a heuristic rule that if a foreign concept (partially) matched with a *majority* of children of a concept, then this concept is a better mapping than (and thus subsumes) its children. The other takes the Bayesian approach that considers the best mapping being the concept that is the lowest in the hierarchy and with the posterior probability greater than 0.5. Details of these two algorithms are given in the next section.

Preliminary experiments, which combine computer simulation and human verification, are described in Section Three. We conclude by discussing issues and future research in Section Four.

2 Algorithms

Let us call the two topic ontologies A and B. Each ontology is a classification hierarchy, with each concept represented as a node in the corresponding

*This research was supported in part by DARPA contract F30602-97-1-0215. This is a student paper. Please send all correspondence to: Sushama Prasad, 4777 Belwood Green, Baltimore, MD 21227.

[1] http://www.daml.org/dl/

[2] http://www-2.cs.cmu.edu/ mccallum/bow/rainbow/

tree. Each node in each hierarchy $(A_1, A_2, \ldots, A_m)$, $(B_1, B_2, \ldots, B_n)$ has a set of exemplar documents (a training set to build its model) that have already been classified as being associated with that node.

The Rainbow classifier is used to compute two raw topic similarity matrices $SMab(A_i, B_j)$ and $SMba(A_i, B_j)$, for each pair of nodes, one from ontology A and one from ontology B. So, $SMab$ is a matrix obtained by classifying the text of A using the model built using the text of B, and vice versa for $SMba$. Let $text(i)$ be the string of all text associated with node i.

$$SMab(A_i, B_j) = Sb(text(A_i), B_j) \quad (1)$$
$$SMba(A_i, B_j) = Sa(text(B_j), A_i). \quad (2)$$

2.1 Simple heuristic approach

This approach realizes the subsumption based on the majority rule. It considers the percentage of children of a node that agree on a mapping to a particular node in the other hierarchy. This percentage, called propagation threshold, can be varied. For each node in the tree, the mappings indicated by the children of the node are examined. The percentage of children that indicate mappings (with non-zero values) to a particular node in the second tree is calculated. If this percentage is greater than or equal to the threshold specified by the user, these mappings and the values associated with them are propagated up to (and thus subsumed by) their parent node. Otherwise, no decision can be made about the parent node, and nothing is propagated. For example, consider a node A with children $(A_1, A_2, \ldots, A_{10})$. Suppose the propagation threshold is set to 60%. So, if children A_1, A_2 andA_3 map to B_1, A_4andA_5 map to B_2, and the other children map to different nodes in B, then no decision can be made for the node A. If, instead, at least 6 children mapped to B_1 with non- zero values, then it could be concluded that A also maps to B_1.

2.2 Bayesian approach

First, consider any non-leaf node, say, N in a hierarchy. Exemplars associated with N are documents that belong to this class but cannot be classified into any one of its subclasses. Therefore, we create one leaf node, called "N-other", as a child of N, and move all exemplars of N to N-other. With this arrangement, raw scores given by Rainbow classifier now become similarity scores between leaves of these two ontologies. Two assumptions are then made:

Assumption 1: all leaves of a hierarchy form a mutually exclusive and exhaustive set.

Assumption 2: the raw score returned by Rainbow classifier $SMba(A_j, B_i)$ is interpreted as $P(A_j \mid B_i)$.

Assumption 1 implies that all children of a node are also mutually exclusive. Assumption 2 allows us to obtain $P(A_j \mid B_i)$ if B_i is a leaf in hierarchy B[3]. When B_i is a non-leaf node, as a superclass, its exemplar documents should include all exemplars associated with all of its subclasses. Therefore, the probability of a leaf node A_j, given a non-leaf node B_i, is

$$\begin{aligned} P(A_j \mid B_i) &= P(A_j \mid \vee_k B_k) \; \forall \, B_k \in children(B_i) \\ &= \sum_{B_k \in B_i} P(A_j \mid B_k) . \frac{P(B_k)}{P(B_i)} . \quad (3) \end{aligned}$$

When specific $P(B_k)/P(B_i)$ is not available, we use a heuristic approximation:

$$P(A_j \mid B_i) \approx \frac{1}{|child(B_i)|} . \sum_{B_k \in B_i} P(A_j \mid B_k) \quad (4)$$

Definition: The concept $A*$ in A said to be the best mapping of a concept B_i in B if

1) $P(A * |B_i) > 0.5$, and

2) none of $A*$'s children A_k has $P(A_k|B_i) > 0.5$.

Condition 1 is used to circumvent the problem of mappings to overly general concepts by going too high on the target hierarchy. This would occur if only relying on $P(A_j|B_i)$ because the posterior probability of any node is the sum of its children's (and the probability of A_{root} is always 1). The value 0.5 is somewhat arbitrary, but it at least indicates that $A*$ is more similar to B_i than not. Condition 2 ensures $A*$ is the most specific concept satisfying condition 1. They together give $A*$ the flavor of the most specific subsumption in description logics. It can be easily

[3] If needed, we normalize these $P(A_j \mid B_i)$ for all j so that they add up to 1.

shown that there is one and only one $A*$ for a given B_i.

The procedure of finding $A*$ consists of a bottom-up step (to compute probabilities of non-leaves) and a top-down step (to identify $A*$).

Bottom-Up:

1. For each leaf node A_j, obtain $P(A_j|B_i)$, either directly from $SMba$ if B_i is a leaf or computed from $SMba$ by Eq. 4 if not.

2. For each non-leaf node A_j, compute

$$P(A_j \mid B_i) = \sum_{A_k \in child(A_j)} P(A_k \mid B_i) \qquad (5)$$

Top-Down:

```
1. set *current* to $A_{root}$.
2. **while** *current* has a child with $P > 0.5$
   set *current* to its most probable child
3. **return** *current*.
```

3 Experiments and results

We have conducted some preliminary experiments in which the automated mapping procedure was performed on the two topic ontologies for a set of selected concepts. Three propagation thresholds (40%, 60%, and 80%) were experimented with the heuristic algorithm. For both algorithms, the resulting mappings were ranked by their respective final scores or probabilities, and were given to five people knowledgeable about computer science for evaluation. Each person was asked to indicate which of the mappings he/she considered to be appropriate. Those mappings that 4 out of 5 survey participants agreed upon were taken to be acceptable. The results were manually analyzed to get an idea of how different people view topics to be related, and thereby judge how accurate the automatically generated mappings are.

Running the heuristic algorithm with threshold of 80% gives the best results of the three thresholds used in the testing. For the top 5%, 10%, 15%, and 20% ranked mappings, the acceptable rates (according to human evaluators) for the heuristic algorithm are 0.8, 0.55, 0.4, 0.4, respectively. The probabilistic algorithm gives better results than the simple heuristic. The corresponding acceptable rates were 0.8, 0.7, 0.68, 0.65, respectively. The better performance of the probabilistic algorithm is probably due to the fact that it has a much stricter constraint on which mappings the system should consider to be good.

Several factors may affect the mapping results. The first is the quality and amount of descriptive text associated with the concepts in each ontology. Most documents associated with our ontologies are abstracts of technical papers taken from ACM's digital library[4] and Citeseer[5]. The problem arises when a document related to databases also talks about other topics. Since the classifier only has knowledge of statistics obtained from the training documents, it may classify this document into concepts of the other ontology which may reference database issues, but are primarily about hardware or computer system implementation. This leads to some inaccuracy when Rainbow builds the models for the ontologies and calculating raw similarity scores. The accuracy may be improved by including a greater number of abstracts associated wit each concept, or including full-length papers rather than short abstracts.

The second factor is the text classifier used. Classification accuracy of a classifier depends on the text classification method it uses and how well this method suits the particular problem. We have used Rainbow in our experiments. But if other classifiers are used, we may have different, possibly better results.

Thirdly, different human evaluators may have different views of the subject, based a different level of knowledge and experience in the field. For example, some persons may agree to a mapping between "ACMTopic/Software/Programming_Techniques" and "ITTopic/Software/Databases", while others may not. This problem may be eased to a degree by including more evaluators.

The last issue to consider is the mutually exclusive assumption we made for the Bayesian approach. This may not hold for all leaf nodes, thus contributing to possible mis-classification.

[4] http://www.acm.org/dl/

[5] http://citeseer.nj.nec.com/

4 Discussion and future work

An attempt has been made to provide solutions for mapping between concepts belonging to two ontologies, using exemplar texts associated with each concept. Our approach is a combination of IR based text classification and Bayesian inference. The values returned by the text classifier are raw numbers. The algorithms we have proposed attempt to make sense of these numbers, and try to produce possible mappings for the user's perusal. Our experiments, though limited in scope, produced encouraging results.

A great number of proposals have been made in the general area of ontology mapping with different approaches [6, 3, 1, 4]. The one that is closest to ours is that of Lacher and Groh [1]. This work also uses documents as explicit information associated with each concept and uses the *Bow* toolkit for the classification task. This approach also treats the scores returned by the text classifier as probabilities. It differs from ours in how these scores are used in determine the final mapping. In their work, only the two most probable nodes that could match a node in the ontology are considered, and the process proceeds to look at their parents if they do not share a common parent.

Another related work is Anchor-PROMPT (Noy, Musen [3]). It takes as input a set of anchors – pairs of related terms defined by the user or automatically identified by lexical matching. The algorithm treats an ontology as a graph with classes as nodes and slots as links. It analyzes the paths in the subgraph limited by the anchors and determines which classes frequently appear in similar positions on similar paths. These classes are likely to represent semantically similar concepts.

We have developed an interface that allows a user to manually select a class from each hierarchy (a *landmark*) and specify a relation between these two classes (e.g. broader, narrower, similar, etc.) thus creating a mapping with special semantics. Similar to anchors, the mappings between landmarks, if properly used, can significantly improve both accuracy and efficiency of concept mapping. Another potentially valuable information source is the set of properties one class may have, because similar concepts not only share similar texts but also similar properties. How to incorporate these and other additional information sources into our automatic mapping framework is one important direction of future research. This would require re-examination of the probabilistic assumptions we have made and development of new algorithms.

Other research directions under active consideration include experimenting and assessing different text classifiers; exploring possible application of our approach in other applications; adaptation of existing mappings when new evidence (e.g., new exemplars) is collected; improving GUI and developing additional tools, to mention just a few.

References

[1] M. S. Lacher, G. Groh. Facilitating the exchange of explicit knowledge through ontology mappings. In *Proceedings of FLAIRS 2001. AAAI 2001.*

[2] H.S. Pinto. Some issues on ontology integration. In *IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5)*, 1999.

[3] N.F. Noy and M.A. Musen. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. *Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*, Seattle, WA, 2001.

[4] H. Wache, T. Vőgele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hbner. Ontology-based integration of information - a survey of existing approaches. *Submitted to IJCAI 2001 Workshop: Ontologies and Information Sharing.*

[5] F. Weisman, N. Roos and P. Vogt. Automatic Ontology Mapping for Agent Communication. *MERIT-Infonomics Research Memorandum series.* (http://meritbbs.unimaas.nl/)

[6] P.C. Weinstein and W.P. Birmingham. Agent communication with differentiated ontologies: eight new measures of description compatibility. Technical Report CSE-TR-383-99, 7, 1999.