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
Baltimore, MD 21250
email: {sprasa2, ypeng, £nin}@csee.umbc.edu

Abstract

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 de£ned in a shared ontology is an oversimpli£ed 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(Weisman, Roos, & Vogt), Pinto(Pinto, Prez, & Martins 1999)). In this paper, we present our work along this direction. This work combines the recently emerging semantic markup language DAML+OIL (for ontology speci£cation), the information retrieval technique (for similarity information collection), and Bayesian reasoning (for similarity synthesis and £nal mapping selection), to provide ontology mapping between two classi£cation hierarchies.

Introduction

Agent technology is one of the most promising ways of distributing and gathering information. In a multi-agent system, an ontology is the basis for communication. The way an agent internally stores information is not known to the environment. Each agent may have its own ontology to organize its data. Mapping one ontology onto another basically means that for each concept in ontology A, a corresponding concept node in ontology B with the same or similar semantics has to be found. Therefore, there is a need to £nd a mapping between the concepts of two ontologies, using either explicit or implicit information. In our work, we have used explict information in the form of documents assigned to each concept in an ontology. We consider classi£cation information for each document and suggest two approaches that use this information to propose a set of possible mappings between the given ontologies. The classi£cation information is in the form of either similarity measures or probability values for single concept nodes.

The two hierarchies we used as examples are ACM topic ontology and a small ITTopic topic ontology which organizes classes of IT related talks in a way different from ACM classi£cation. 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.

An example of a possible application is in ITTALKS.org (Finin), which is a web-based portal developed at UMBC to provide intelligent noti£cation of talks. The system is agent-based, and thus each agent may have its own ontology. Algorithms for mapping between different agents' ontologies would enable "better" noti£cations.

Previous related work

A great number of proposals have been made in the general area of ontology mapping with different approaches (Weinstein & Birmingham 1999; Noy & Musen 2001; Madhavan, Bernstein, & Rahm 2001; Weinstein & Birmingham 1999; Mitra, Wiederhold, & Jannink). One such work that is very similar to ours is that of Lacher and Groh (Lacher & Groh 2001). This work also uses documents as explicit information associated with each concept and uses the *Bow* toolkit for the classi£cation task. It also treats the scores returned by the text classi£er as probabilities. The difference between this approach and ours is in how these scores are used in determine the £nal mappings. 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 (Noy & Musen 2001)). It takes as input a set of anchors – pairs of related terms de£ned by the user or automatically identi£ed 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.

Overview

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 Rain-

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

bow text classi£er[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 classi£er. The raw similarity scores returned by the classi£er are used by the mapper to produce a set of possible mappings between the two ontologies. Fig. 1 shows the system components and the ¤ow of inputs to the components.

Based on the subsumption operation in description logics, two algorithms have been developed to synthesize the raw similarity scores toward the £nal 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.

We discuss preliminary experiments, which combine computer simulation and human veri£cation. We conclude by discussing issues and future research.

Algorithms

Let us call the two topic ontologies A and B. Each ontology is a classi£cation hierarchy, with each concept represented as a node in the corresponding 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 classi£ed as being associated with that node.

The Rainbow classi£er 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.

$$SMab(A_i, B_j) = Sb(text(A_i), B_j) \quad (1)$$

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

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 speci£ed 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]http://www-2.cs.cmu.edu/ mccallum/bow/rainbow/

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 classi£ed 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 classi£er 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 classi£er $SMba(A_j, B_i)$ 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

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

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

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

De£nition: 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 speci£c concept satisfying condition 1. They together give $A*$ the ¤avor of the most speci£c subsumption in description logics. It can be easily shown that there is one and only one $A*$ for a given B_i.

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

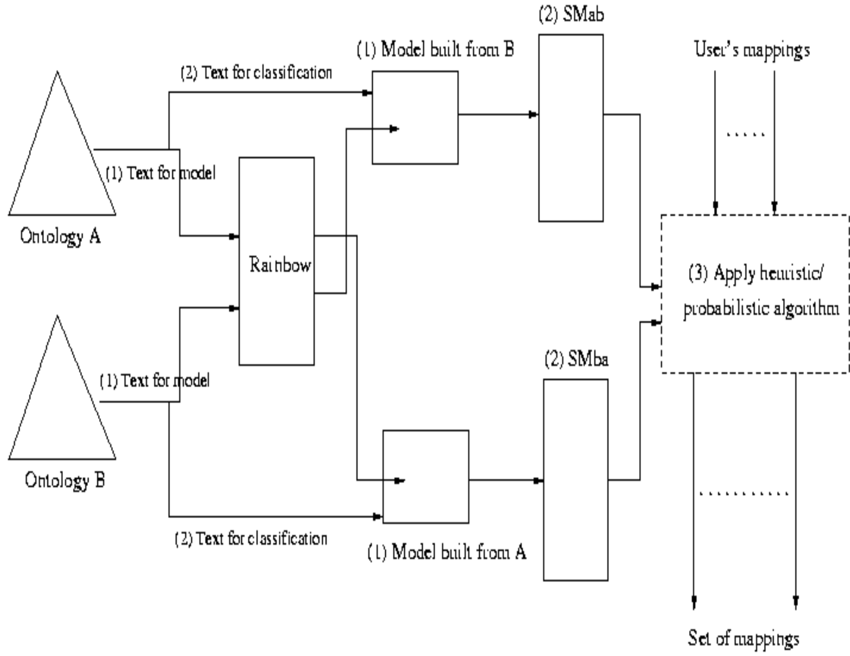

Figure 1: System components and ¤ow

The procedure of £nding $A*$ 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*.

User Input

We have developed a prototype GUI to aid the user in the manual mapping process. The user can select a node from each tree and specify a mapping (*landmark*). When specifying *landmarks*, a property can be assigned to the link - Broader, Similar, or Narrower. These properties, if used properly, can signi£cantly improve both accuracy and ef£ciency of the concept mapping. Once the user has £nished specifying landmark categories, he can select one of the two approaches for classi£cation. From then on, the automated mapper takes over, considering these mappings to be absolute. Fig. 2 shows a snapshot of the GUI.

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 £nal scores or probabilities, and were given to £ve 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. Fig. 3 shows the results obtained for the probabilistic algorithm.

Several factors may affect the mapping results. The £rst 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

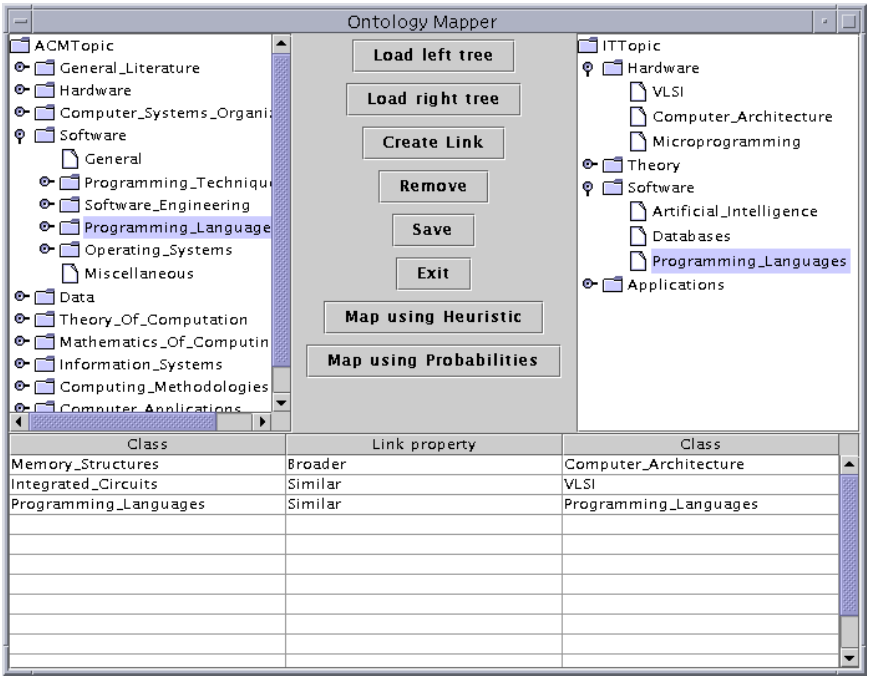

Figure 2: GUI for specifying manual mappings

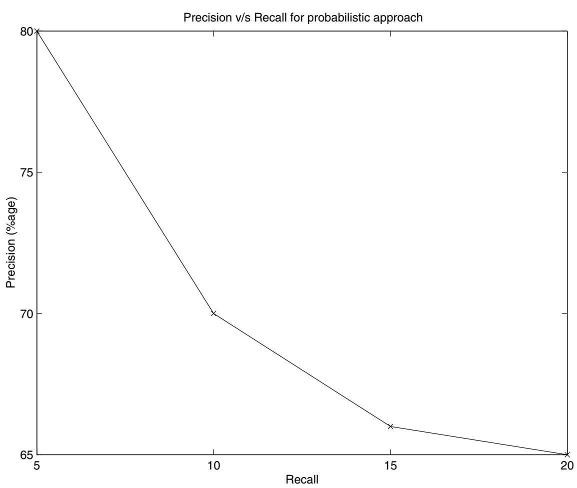

Figure 3: Probabilistic algorithm results

ACM's digital library[4] and Citeseer[5]. The problem arises when a document related to databases also talks about other topics. Since the classi£er 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 classi£er used. Classi£cation accuracy of a classi£er depends on the text classi£cation method it uses and how well this method suits the particular problem. We have used Rainbow in our experiments. But if other classi£ers are used, we may have different, possibly better results.

Thirdly, different human evaluators may have different views of the subject, based on different levels of knowledge and experience in the £eld. 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 misclassi£cation.

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 classi£cation and Bayesian inference. The values returned by the text classi£er are raw numbers. The algorithms we have proposed attempt to make sense of these numbers, and try to produce possible map-

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

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

pings for the user's perusal. Our experiments, though limited in scope, produced encouraging results.

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 signi£cantly improve both accuracy and ef£ciency 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 with and assessing different text classi£ers; exploring possible application of our approach in other applications; adaptation of existing mappings when new evidence (e.g., new exemplars) is collected; improving the GUI and developing additional tools, to mention just a few.

Acknowledgement

This research was supported in part by DARPA contract F30602-97-1-0215.

References

Finin, T., e. a. In *International Semantic Web Working Symposium (SWWS), Stanford University, California, USA*.

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

Madhavan, J.; Bernstein, P.; and Rahm, E. 2001. Generic schema matching with cupid. In *Proceedings of the 27th VLDB Conference, Roma, Italy*.

Mitra, P.; Wiederhold, G.; and Jannink, J. Semi-automatic integration of knowledge sources.

Noy, N., and Musen, M. 2001. Anchor-prompt: Using non-local context for semantic matching. In *Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Arti£cial Intelligence (IJCAI-2001)*.

Pinto, H.; Prez, A.; and Martins, J. 1999. Some issues on ontology integration. In *Workshop on Ontologies and Problem-solving Methods*. IJCAI.

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

Weisman, F.; Roos, N.; and Vogt, P. Automatic ontology mapping for agent communication. In *MERIT-Infonomics Research Memorandum series*.