Algorithms for Ontological Mediation

Alistair E. Campbell[1] and Stuart C. Shapiro[2]
[1,2]Department of Computer Science
And [2]Center for Cognitive Science
State University of New York at Buffalo
226 Bell Hall, Box 602000
Buffalo, New York 14260-2000
aec@cs.buffalo.edu, shapiro@cs.buffalo.edu

Abstract

We lay the foundation for ontological mediation as a method for resolving communication difficulties resulting from different ontologies. The notion of hierarchical relations enables a theory of orientation or direction of ontologies to be presented. We describe an ontologcial mediator as being able to think about (or conceptualize) concepts from ontologies and find equivalences between them. Algorithms for finding the meanings of unfamiliar words by asking questions are introduced and evaluated experimentally.

Introduction

Clearly, in order for communication between computational agents to be truly successful, each agent must be able to understand what the other says. Presently, this involves deciding ahead of time on the following:

I. a syntax and semantics for the language in which they communicate (a popular one is KIF (Genesereth 1995)), and

II. an ontology, or domain conceptualization that sets forth the terminology they may use, along with relations that hold between the concepts that these terms denote.

One way to make sure that both of these things happen is to develop a single ontology with a single set of terms for each domain, and require that all communicating parties use only that ontology in their dialogue. We call this the *single ontology proposal*.

However, the reality is that various agents can and often do use different terms to denote elements in a common domain, and this presents a pervasive problem: Words that are not in one agent's ontology will be completely unintelligible when presented by another agent, even if they have agreed on a common language of communication (an *interlingua*) ahead of time, and even if their ontologies are similar, even significantly overlapping.

This problem often occurs because the agents' ontologies are designed for different purposes. We should reject the single ontology proposal because it is impossible to implement: even the designers of the ontologies themselves cannot agree on terminology. Worse yet, they often cannot agree on a taxonomization of the domain into represented concepts. For example, notice the differences between upper levels of the CYC (Lenat & Guha 1990; Lenat 1995), and Penman (Bateman *et al.* 1990) ontologies shown in figure 1.

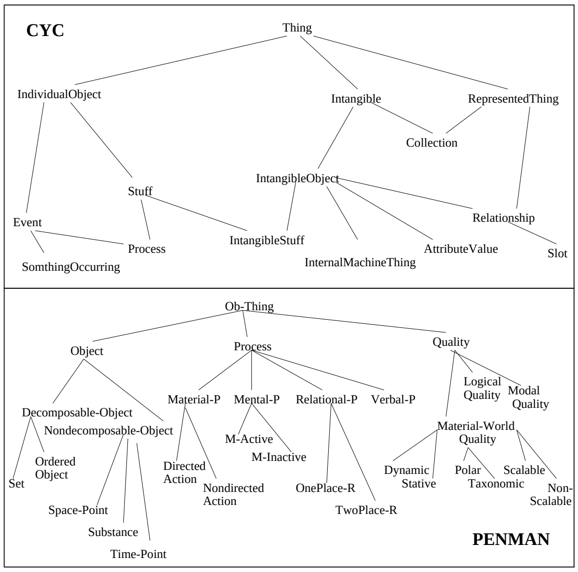

Figure 1: CYC and Penman Upper Levels

Moreover, useful knowledge resources designed before a standard ontology is adopted will not be able to participate in information interchange without the assistance of some sort of translator or mediator to facilitate dialogue with other agents. Since these tools are expensive to develop and maintain, this effectively eliminates legacy systems as competitive knowledge resources at large.

Also, without unacceptable forced compliance to *the standard ontologies*, anyone can create new and potentially useful knowledge agents with which communication is impossible even if they do use some conventional language and communication protocols.

Instead we advocate an approach where agent designers are free to use whatever ontology makes sense for them, and when problems of terminology arise, they are solved by an *ontological mediator*.

Vocabulary

Let's suppose that agent A and agent B want to communicate about some domain. They have decided on an interlingua, a common communication language, and each has adopted an ontology, or domain conceptualization. This means that they have an established vocabulary from which neither may stray.

But how crucial is it that both agents have *exactly the same* vocabulary? People don't have exactly the same vocabulary, yet we communicate very well most of the time. When misunderstandings occur, they are often easily cleared up. Legacy systems and most current knowledge resources are incapable of clearing up miscommunications because they lack the intelligence to do so. Work toward giving information agents this capability is progressing, but in the interim, machines can't communicate.

Mediation

One promising approach to this problem is to build specialized agents which facilitate communication between communicants who have adopted different ontologies, or even no formal ontology at all. Indeed, given that agents have adopted an interlingua and communication protocol, they can *try* to communicate. The mediator then tries to repair miscommunications as they occur.

We are concerned not with the process of detecting misunderstandings, but rather with ways to resolve communication problems. We focus on the problem of agents having different vocabularies. In that context, it is possible for a speaker (S) to use a word (W) unfamiliar to a listener (L).

Mediator

We have designed an *ontological mediator*, an agent capable of reasoning about the ontologies of two communicating agents, or *communicants*, learning about what W means for S, and looking for an ontological translation (W') that means for L, the same thing in the domain that W means for S.

Fundamentals

Before proceeding with a discussion of algorithms for ontological mediation, we first set forth some assumptions and definitions, and make some clarifying remarks.

Common words mean the same thing.

We make the following simplifying assumption:

Rule 1 *If two agents are communicating about the same domain, then if both of them know some word, then it means the same thing to both of them.*

The rationale for this assumption is that when agents are communicating, each implicitly assumes that a word used by the other means the same thing as it does to it. People don't go around wondering whether each word they hear really means what they think it does, and their communication with other people is usually free of error. Of course, this assumption can lead to problems when common words really don't mean the same thing. Then it becomes the agents' duty to detect miscommunication. Work is being done in this area (see, for example (McRoy 1996)) but this is not the focus of our current research. We are more concerned with using mediation techniques to find correspondences between concepts in ontologies. This presupposes detection, since the agents have called a mediator to help them.

Ontologies

The word "ontology" is used by many researchers to mean a variety of similar but distinct things. Without making a strong or precise statement as to what ontologies should be necessarily, we present some issues with respect to ontologies that our research addresses.

Words vs. Concepts Contrary to many ontology designers, who do not seem to distinguish between word (or symbol) and concept, we take an ontology to be an organization of an agent's *concepts* by some set of ontological relations. A concept is a particular agent's conceptualization of an element of the domain of discourse, and each concept can be denoted by one or more words. This way, words can be shared between agents, but concepts cannot. Naturally, we require a mapping between words and concepts to support reasoning about agents' concepts. For a given agent, we currently assume a 1-1, onto mapping between concepts and words. Presently, we do not have algorithms that give a proper treatment of polysemy or synonomy of words for ontological mediation.

Concepts If an ontological mediator is to find words in one ontology that have the same meaning as words in another ontology, the mediator must be thinking about the concepts in those ontologies. The notion of a "concept" is very slippery, and frequently means different things to different people. Therefore, for the purpose of describing these algorithms and their underlying theory, we make the following definitions.

1. For any agent A and domain element O, if A knows about or can think about O, then there exists a mental representation C in A's mind, which *represents* O. We write $[\![C]\!]_A = O$.

2. *Concept:* The mental entity C which exists in the mind of an agent and serves to represent some domain element for that agent.

3. *OM-Concept:* The mental entity C' which exists in the mind of the ontological mediator that is thinking about C, that is, thinking about some concept in the mind of another agent, and how that concept might fit into the agent's ontology.

Note one important implication of the distinction: The "domain" of thought for an ontological mediator is not the same as the communicants' domain. Rather, the OM's domain is that of *concepts* in the communicants' ontologies. While the communicants are "thinking about" elements of their own domain, the OM is thinking about those concepts invoked by the communicant's thinking. Thus, whenever agent A uses a word W, it expresses some concept C, which in turn represents some domain entity O for A. Therefore, the first time OM hears A use W, OM builds in its own mind an om-concept C' to represent that concept. Hence $[\![\text{C'}]\!]_{OM} = C$, and of course $[\![\text{C}]\!]_A = O$.

Ontological Relations

An ontological relation is simply any relation commonly used in the organization of ontologies. Whether a relation is truly ontological is a matter of opinion, but for example, some kind of **subclass/superclass** relation pair is almost always used to form a taxonomic hierarchy.

Hierarchical generalizers, and specializers A hierarchical ontological relation is any ontological relation that organizes concepts into a hierarchy, taxonomy, or similar structure. Hierarchical relations are related to but distinct from transitive relations. For example, the transitive relation **ancestor** is related to the hierarchical relation **parent**.

The hierarchical ontological relations are important for ontological mediation because they form the hierarchies organizing the concepts in the ontology. When a relation is hierarchical, we can think of it as having an direction or orientation, either as a *generalizer*, relating a concept to concepts above it (e.g., its "superconcepts"), and moving "up" the hierarchy, or as a *specializer*, relating a concept to concepts below it (its "subconcepts"), and moving "down". For example, directSuperClass is a hierarchical generalizer, while directSubClass is a hierarchical specializer.

The "up" and "down" directions are merely conventions, of course, in that they relate to the way we tend to draw pictures of hierarchies as trees. We start at some root concept or concepts and fan out via some hierarchical specializer. How do we know that **directSubClass** is the specializer (down direction) and that **directSuperClass** is the generalizer (up direction)? We expect fan-out with specializers, that is, specializers tend to relate several subconcepts to a single superconcepts. For a pair of hierarchical relations R and R' (the converse of R), we examine the sets of concepts $X = \{x|\exists y R(x,y)\}$ and $Y = \{y|\exists x R(x,y)\}$. If $|Y| > |X|$ then R is a specializer, otherwise R is a generalizer.

If R is a hierarchical relation, then R' is its converse, i.e., $R(C_1, C_2) \equiv R'(C_2, C_1)$. It follows naturally that if R is a generalizer, then R' is a specializer, and vice versa.

We say that a concept P is a "parent" (with respect to R) of another concept C if $R(C, P)$ for some hierarchical generalizer R. Likewise, we say that a concept C is a "child" of P if $R(P, C)$ for some hierarchical specializer R.

Relation notation

By convention, $R(X, Y)$ means that Y bears the R relation to X, for example, we say $subclass(animal, dog)$ to mean that *dog* is a subclass of *animal*. We choose this convention to reflect the question-asking approach where questions are asked of the domain and answers are given in the range. For example, in "What are the subclasses of animal?" we have the question in terms of a relation: $subclass(animal, ?x)$, or functionally, as in $subclass(animal) = ?x$.

Tangled Hierarchies

For many ontologies, the taxonomic hierarchy is structured as a tree (or as a forest), where any given concept can have at most one superconcept. Other ontologies can be tangled hierarchies with multiple inheritance. The techniques of ontological mediation presented here *do* allow for mediation with tangled hierarchies.

Algorithms

In this section, we discuss various algorithms for ontological mediation. We define $word(C, A)$ to be the word that agent A uses to express concept C, and *concept*(W, A) to be the om-concept representing the concept that W expresses for A, if one exists, undefined otherwise. Also, let *knows*(A, W) be true if and only if *concept*(W, A) is defined, false otherwise.

We define the following operations:

- *Ontology*(A) : return the set of om-concepts that OM currently uses to represent concepts in A's ontology.

- *Agent*(C) : returns a representation of the agent that C is an om-concept for. This representation is used to direct questions to the agent.

The following algorithm exists in support of ontological mediation algorithms by asking questions of the communicants as needed to establish OM's knowledge of ontological relationships. **Evaluate** takes a relation R, and an om-concept C, and returns a set of om-concepts such that Agent(C) believes $R([\![C]\!]_{Agent(C)}, [\![C']\!]_{Agent(C)})$ for each om-concept C' in the set. Results are cached so that multiple calls to evaluate the same question do not result in multiple queries issued.

```
Algorithm Evaluate(R,C): set of om-concept
let A ← Agent(C)
Build a query Q in A's interlingua to
ask ''What bears relation R to
word(C,Agent(C))?''
Issue Q to Agent(C). The response to
the query will be a set of words S.
let Answer ← {}
for V ∈ S do
  assert R(C,concept(V,A))
  let Answer ← Anwswer + concept(V,A)
end for
return Answer
```

The first two algorithms below each take as arguments a word W used by agent S and not known by agent L, and return a set of om-concepts representing possible ontological translations. More formally, when X is the om-concept for which $word([\![X]\!]_{OM}, S) = W$, given any om-concept Y in the set returned by the algorithm, there is reason to believe that $[\![[\![X]\!]_{OM}]\!]_S = [\![[\![Y]\!]_{OM}]\!]_L$.

Recursive over one relation (MedTax)

The first algorithm explores an ontology along one hierarchical relation, given by parameter R. It is called $MedTax$ because an obvious choice for R is either *SubClass* or *SuperClass*, which will result in exploration of the taxonomic hierarchies of the ontologies.

```
Algorithm MedTax (W,S,L,R): set of
                            om-concept
let Q ← {}
for P ∈ Evaluate(R,concept(W,S) do
  if knows(L,word(P,S)) then
    let Q ← Q + concept(word(P,S),L)
  else
    let Q ← Q ∪  MedTax(word(P,S),S,L,R)
  end if
end for
F ← {}
for P ∈ Q do
  for C ∈ Evaluate(R′,P) do
    if not knows(S,word(C,L) then
      F ← F + C
    end if
  end for
end for
return F
```

Multiple relations(MedOnt)

We can extend this algorithm to handle multiple hierarchical ontological relations, such as Part/Whole. Now, each hierarchical ontological relation forms its own hierarchy in which the unknown word is situated in the listener's ontology.

Again, we find the translation of a word used by S but unknown to L by starting at the unknown word in the speaker's ontology, then crawling up (or down) the hierarchies of the speaker to points where ontological translations of the word at those points has been made already, (or is easy to make immediately because the listener knows the word) then crawl back down (or up) the listener's hierarchies.

```
Algorithm MedOnt (W,S,L):
          set of om-concept
let G ← {}
for each relation
R ∈ HierarchicalRelations do
  let G ← G ∪ MedTax(W,S,L,R)
end for
return G
```

Note that **MedOnt** is a union-forming algorithm, rather than an intersection-forming one. That is, it returns om-concepts that are found by exploring via one or more hierarchical relations, rather than restricted to having been found through every relation. It returns a set of candidates for ontological translation, and does not calculate which is the best one.

Choosing the best candidate (MedCount)

This algorithm, unlike the previous algorithms, returns a pair: (1) the single om-concept representing the listener's concept which the mediator believes to be equivalent to the speaker's concept expressed by an unknown word W, and (2) a measure of the mediator's confidence in this ontological translation.

We introduce the notation $A \equiv_Y B$ to mean that concept A is known by OM to be equivalent to concept B with confidence measure Y.

```
Algorithm:  MedCount(W,S,L):
            om-concept × Real
1.  if knows(L,W) then
2.    return (concept(W,L),1)
3.  end if
4.  if concept(W,S) ≡Y X then
5.    return (X,Y)
6.  end if
7.  let AllCandidates ← {}
8.  for R ∈ HierarchicalRelations do
9.    let Candidates ← MedTax(W,S,L,R)
10.   let CandidatesByRelations ←
      CandidatesByRelations + Candidates
11.   let AllCandidates ←
      AllCandidates ∪ Candidates
12. end for
13  choose C ∈ AllCandidates such that the
    number of sets in CandidatesByRelations
    that contain C is maximized.
14  let Y ← the number of sets in
    which C occurs.
15. assert concept(W,S) ≡Y C
16. return (C,Y)
```

Experiments with WordNet

The WordNet (Miller *et al.* 1993; Miller 1995) lexical ontology organizes concepts called "synsets," which are sets of words considered synonymous in a certain context. It also contains other information about the role of words in the English language, including verb usage, synonomy, and relevance to situations. Primarily we are interested in some of WordNet's hierarchies, including the taxonomic hierarchy.

Variables

Since WordNet is such a large ontology, we controlled two independent binary variables in the experiment, **Synonyms**, and **AllowAllSenses**. These are explained below.

Synonyms One approach to WordNet is to consider each synset as a separate mental concept in the mind of the agent who uses WordNet as its ontology. When the agent expresses that concept, he uses one or more of the words in the synset. If so the agent supports *synonomy*. However, deciding which synonym to use is difficult to say the least, and may be a reason why many if not most ontologies don't support synonomy.

AllowAllSenses The agent playing the role of WordNet receives queries from the ontological mediator, then in turn makes an appropriate access to its WordNet component. Each query returns a sequence of zero or more groups of output, one for each relevant synset the word was in. If **AllowAllSenses** was not set, the agent only reported the information from the first block, ignoring the others. Conversely, if **AllowAllSenses** was set, then the agent reported information from all synsets.

Experiment

We devised two agents, appropriately named "AMERICAN" and "BRITISH" because they were constructed to use the corresponding dialect of the English language. Both agents use the WordNet ontology, but are restricted from using words strictly from the other's dialect (they pretend not to know them). The dialect restrictions come from the Cambridge Encyclopedia of the English Language, (Crystal 1995, p. 309). Naturally we only used word pairs where both words exist in WordNet in the same synset. We chose 57 word pairs where both words were present in WordNet and members of the same snyset, for example, `(lift,elevator)`, `(patience, solitaire)`, `(holiday, vacation)`, `(draughts, checkers)`.

We then tested the `MedCount` algorithm mediating from an American speaker to a British listener, and then vice versa from a British speaker to an American listener. There were four Hierarchical relations used: **SubClass, Superclass, PartOf,** and **HasPart**.

When the mediator returns the correct word from the word pair, that is called a *success*. When the mediator returns some other word, that is called an *error*, and when the mediator can not find any word for an ontological translation that is called a *miss*.

Sen	Syn	Suc	Err	Mis	Rat	Cer	CPU
Off	Off	28	2	27	49%	.85	0.97s
Off	On	33	3	21	58%	.79	2.40s
On	Off	39	5	13	68%	.82	3.03s
On	On	40	7	10	70%	.85	6.82s

Table 1: British Speaker/American Listener

Table 1 summarizes the performance of the `MedCount` algorithm under combinations of Al-

Sen	Syn	Suc	Err	Mis	Rat	Cer	CPU
Off	Off	19	2	36	33%	.85	1.03s
Off	On	35	3	19	61%	.78	2.38s
On	Off	4	7	46	7%	.81	2.20s
On	On	42	4	11	74%	.82	5.22s

Table 2: American Speaker/British Listener

lowAllSenses (Sen) and Synonyms (Syn), showing the numbers of successes, errors, misses, success rate ($Success/57 \times 100\%$) , the average certainty over all successes (Cer), and average CPU time, when the speaker is "BRITISH" and the listener is "AMERICAN." Table 2 gives the same data for when the speaker is "AMERICAN" and the listener is "BRITISH".

Analysis

The first remarkable difference between an American speaker vs. a British speaker is that the success rate plummets when *Synonyms* is turned off. This reflects a bias in WordNet for putting the American words first in the synsets. If the British word is at the end, it will not be reported when *Synonyms* is on, thus it will not be found, and the miss rate increases.

Another reason for seemingly low success rates even with both *Synonyms* and *AllowAllSenses* on is due to polysemy inherent in WordNet. The mediator can't distinguish among ontology data for multiple word senes, so errors occur when there are too many translation candidates and the wrong one is picked.

Discussion and Future Work

The Ontological Mediator asks appropriate questions of a speaker and listener to find words in the listener's ontology it believes mean the same as words in the speaker's. We have demonstrated that ontological mediation is a promising technique for assisting other agents with communication. After successfully testing algorithms on mostly identical ontologies we are are prepared to proceed to mediation tasks involving agents with greater contrasting ontologies. We expect that since many of the misses and errors are due to WordNet's polysemous nature, performance will improve when dealing with non-polysemous ontologies.

Long response times are due mainly to the size and density of the WordNet ontology. The ontological mediator running `MedCount` must explore a sizable portion of each agent's ontology to arrive at its conclusion. Even though much of this exploration involves common words, OM still must establish many equivalences between om-concepts that are expressed by the same word. Because WordNet is inherently a polysemous ontology, OM must explore several dead ends. For example, in discovering (successfully) that "pushcart" is synonymous with "stroller," OM must look at senses of the word "carriage" which then brings in all the parts of a typewriter. Work on pruning this sort of search is being considered.

Meanwhile we plan to apply ontological mediation algorithms to other ontologies including the Unified Medical Language System (UMLS) (Humphreys & Lindberg 1993). Mediating between two different ontologies, UMLS and WordNet will lead to new ideas for ontological mediation algorithms. Another experiment could involve human subjects, for example, those searching a database and are looking for just the right keyword to find some target. We expect these experiments to lead to more robust ontological mediation algorithms.

Acknowledgements

This research was supported in part by Rome Laboratory and the Defense Advanced Research Projects Agency under USAF contract F30602-93-C-0177.

References

Bateman, J. A.; Kasper, R. T.; Moore, J. D.; and Whitney, R. A. 1990. A general organization of knowledge for natural language processing: The penman upper model. Technical report, USC/Information Sciences Institute.

Crystal, D. 1995. *The Cambridge Encyclopedia of the English Language.* Cambridge University Press.

Genesereth, M. R. 1995. Knowledge Interchange Format. Available at URL: http://logic.stanford.edu/kif.html.

Humphreys, B. L., and Lindberg, D. A. B. 1993. The umls project: Making the conceptual connection between users and the information they need. *Bulletin of the Medical Library Association* 81(2):170.

Lenat, D., and Guha, R. 1990. *Building Large Knowlede-Based Systems: Representation and Inference in the CYC Project.* Addison-Wesley.

Lenat, D. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM* 38(11):33–38.

McRoy, S., ed. 1996. *AAAI-96 Workshop on Detecting, Preventing, and Repairing Human-Machine Miscommunication.*

Miller, G. A.; Beckwith, R.; Fellbaum, C.; Gross, D.; and Miller, K. 1993. Introduction to Word-

Net: An On-line Lexical Database. Available at URL: http://clarity.princeton.edu:80/~wn/.

Miller, G. A. 1995. WordNet: A Lexical Database for English. *Communications of ACM* 38(11):39–41.