ASSIMILATION:
A Strategy for Implementing Self-Reorganizing Knowledge Bases

Jane Terry Nutter
Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24060

Abstract

Assimilation is a process by which a knowledge base restructures itself to improve the organization of and access to information in the base. This paper presents a strategy for implementing assimilation in propositional knowledge bases which distinguish between the axioms of the system's knowledge (called the context) and the derived consequences of those axioms (called the belief space). The strategy in question takes advantage of housekeeping phases in which the system discards accumulated clutter to discover useful patterns of access on the basis of which the context can be reorganized. Unused axioms are replaced by their more useful consequences; derivable generalizations that shorten common inference paths are added to the belief space.

I. Introduction

Systems that use propositional knowledge bases must choose how to organize those bases: what information to make explicit and what information implicit, how to structure the explicit information, how much information to store and how much to infer as needed, and so on. Up to now, these decisions have rested with system designers. Fundamental choices have been made before implementation, in selecting a particular set of propositions to begin with and in decisions such as whether to retain results of inferences (final results, intermediate, or both). Changing these decisions usually involves manual intervention. To change what is explicitly included, the designers go in by hand and take out some propositions, put in others, and so on. To change what information is automatically added and retained, more extensive and costly alterations must be made.

How systems organize their knowledge obviously affects their performance. But the nature of a given domain does not as a rule dictate a single best organization of its information. On the contrary, what organization is best for a given system depends on the circumstances under which the system is to be used, and on the interests and desires of its users. These circumstances, interests, and desires vary from one system to another in a single domain, and even over time for a single system. As a consequence, getting the decisions right at design time requires extensive customizing, if indeed it is possible at all.

The alternative is to get the systems to reorganize their knowledge bases themselves. This course has several evident advantages. It minimizes the impact of errors in initial decisions, it lets systems adapt to the environments in which they are used, and it provides a way for systems to optimize their knowledge bases relative to the inferences they are actually called on to make. The approach reported here hypothesizes that this function can usefully be combined with automated "forgetting": at regular intervals, the system examines the portions of its belief space which have not been accessed recently, discards some of the information as useless, and reorganizes the rest to reflect the way the information has proved useful. This combined process of restructuring and controlled forgetting is called *assimilation*; this paper describes a strategy for assimilating information automatically which contributes toward the above goals at the same time that it lets systems "housekeep" to remove information clutter that is not proving useful. In the process, the system forms useful generalizations on the basis of the patterns of use and discovery of information.

II. Structure of the Underlying System

The strategy proposed here can be adapted to a variety of system architectures, but the discussion will be in terms of a knowledge base implemented in SNeBR [Martins, 1983a] [Martins, 1983b] [Martins and Shapiro, 1983] [Martins and Shapiro, 1984] [Martins and Shapiro, 1986a] [Martins and Shapiro, 1986b] [Martins and Shapiro, 1986c], a semantic network architecture with a relevance-logic style belief revision system, implemented on SNePS [Shapiro, 1979] [Shapiro and Rapaport, 1986] and augmented with a monotonic logic for reasoning with default-style generalizations [Nutter, 1983a] [Nutter, 1983b]. This section describes the aspects of that architecture which contribute significantly to the discussion of assimilation.

Propositions are represented as fragments of network, with logical relations (connectives, quantifiers, etc.) represented by reserved arcs. Since quantification is over nodes, which may represent individuals, properties, propositions, or any other objects of thought, the logic in question is higher order. In addition, the mapping between SNePS representations and propositions in standard higher-order predicate logic is not always one-one. For instance, SNePS relations can take a set (as opposed to a tuple) of arguments. Hence for a symmetric relation R, SNePS can express in a single atomic proposition node the same information as two first order atomic propositions R(a,b) and R(b,a); and the SNePS representation eliminates axioms of symmetry altogether. However, for the purposes here, the SNePS

representation can be treated as equivalent to a representation in standard logic (first or higher order), augmented by a capacity for representing default generalizations.

The default reasoning system introduces a logical operator p, which takes a proposition or formula and marks it as uncertain (p can be read roughly as "presumably"). For the purposes here, the only significant rule about p is that p's are "inherited" through inferences. That is, suppose that a proposition ϕ can be inferred from a set of propositions $\Psi = \{\psi 1,...,\psi n\}$, and suppose that for $1 \leq i \leq n$, $\psi i' = \psi i$ or $\psi i' = p\psi i$. Then from $\Psi' = \{\psi 1',...,\psi n'\}$ the system can infer $p\phi$. For a discussion of the default logic, see [Nutter, 1983a].

The belief revision system introduces the concepts of *contexts* and *belief spaces*. A *context* is a set of propositions taken as hypotheses which form the deductive basis of a belief space, and which may be thought of as the axiom set of a potential agent's beliefs. The *belief space* associated with a given context contains all propositions which have been deduced from that context. The belief space does not automatically include all propositions entailed by the context, since agents are taken as knowing (or believing) only those propositions actually inferred (and those trivially subsumed by them). As propositions are deduced from hypotheses in a context and other members of the associated belief space, those propositions not trivially subsumed by propositions already present are added to the belief space, along with other propositions proved along the way.

Propositions may belong to several belief spaces, i.e., the belief spaces of different contexts can and frequently will overlap. In addition to modeling the system's beliefs and beliefs of other agents, contexts provide settings for hypothetical reasoning, and so on. At any given time, there is a distinguished context called the *current context* (CC), which contains the hypotheses of the system's belief space. This is the context we will be most concerned with here.

Propositions in a belief space have associated deductive histories (one for each deduction which had the proposition as its conclusion). Each deductive history includes a record of the proposition's *set-of-support*, which is the list of hypotheses actually used in deriving the conclusion in that particular deduction (where a previously proved proposition A is used, the set-of-support includes A's set-of-support, not A itself). Belief spaces can be identified using sets-of-support: a proposition is in the belief space of a context C provided that it has at least one deductive history whose set-of-support is a subset of C.

The assimilation strategy proposed here, then, is targeted on systems whose knowledge base can be construed as having the structure given in figure 1. The effect of assimilation will be to alter both the contents and the structure of the CC and its associated belief space.

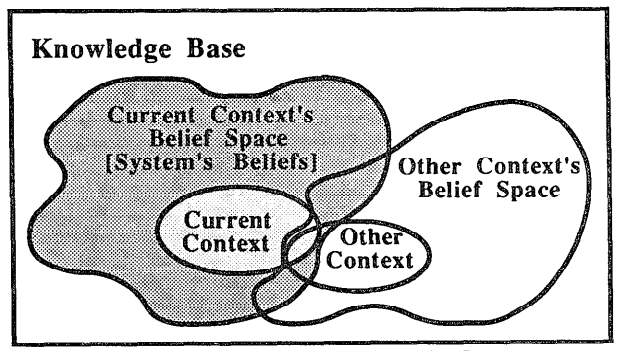

Figure 1. Structure of the Knowledge Base

III. Assimilation and Forgetting

The essential idea behind this approach to assimilation is to make use of internal housekeeping cycles for discarding "clutter". The architecture described above saves not only all results of deductions (answers to top-level user questions, for instance), but also all intermediate results not trivially subsumed by known propositions. For instance, say that the CC contains representations for the propositions $\forall x(\text{Bird}(x) \supset \text{Has-feathers}(x))$, $\forall x(\text{Ostrich}(x) \supset \text{Bird}(x))$ and Ostrich(Oscar), and suppose that the system is asked whether Oscar has feathers. Upon completing the deduction, the system will add a representation of Has-feathers(Oscar) and Bird(Oscar). For questions that require realistic amounts of inference, these intermediate results can increase the size of the knowledge base significantly.

The system could simply choose not to add the intermediate results, but then it will have to repeat the same chains of reasoning, sometimes quite long ones. In other words, the question whether to retain these intermediate results represents a classic time-space trade-off, in this case a choice between reinventing the wheel and remembering everything the system has ever known, however trivial. The ideal answer would be to retain only those results which the system will actually want and find useful in the future. The key idea here is that these may be easier to identify in retrospect than in advance, and identifying them in retrospect can be worthwhile.

Suppose that the system time-stamps every proposition in the knowledge base every time that the proposition is accessed. ("Accessed" here means actually used, not just involved in a set-of-support manipulation, for instance.) Suppose also that a latency time period t is chosen to reflect "recent use". Then at regular intervals, the system can scan for propositions which have not been accessed in the period from now - t to now. One simple rule would be the following: if these propositions do not belong to the CC, then they constitute clutter and can be forgotten.

But not all clutter is useless. For example, suppose that an element of the CC has not been accessed in the latency time. It may be that instead of using that proposition in making deductions, the system is using other members of the belief space which in fact shorten deductive chains. Those other members of the belief space may have been proved using the original hypothesis; it cannot simply be dropped without curtailing the belief space and losing information that is actively in use. But it may be possible to *replace* that hypothesis by one or more of the propositions in the belief space. That is, the fact that an hypothesis

has not been used in a while may show that the current axiomatization is less efficient than an available alternative. This alternative provides a restructuring of the belief space to adapt to the system's environment.

Even unaccessed propositions in the belief space but outside the CC may not be simple clutter. For instance, suppose that the system has unaccessed propositions of the form f(a) for different instances of a. If the system does not contain $\forall x f(x)$, and if that is true, maybe it should add it. That is, instances of concrete propositions of the same form may reflect frequently followed paths of inference which can be shortened.

So to optimize the balance of time wasted on repeated inferences versus space wasted on useless results, the system performs controlled forgetting, checking first that no crucial information will be lost. But before simply forgetting any single item, the system reflects on it to see whether it indicates a useful restructuring or other adaptation of the belief space. It is this reflection and consequent altering and restructuring which constitute assimilation.

IV. The Assimilation Strategy

Select a latency time *t* to represent the longest period for which a proposition may be unused without being considered for assimilation or forgetting. On each assimilation cycle, the system scans through the knowledge base to form the set $\phi = \{\phi 1,\ldots,\phi n \mid$ for $1 \leq i \leq n$, ϕi has not been accessed for at least *t* long$\}$; propositions in ϕ are called stale. Because the strategy is sensitive to the order in which the members of ϕ are considered, they are ordered with the most recently accessed last. That is, for $i < j$, time-of-last-access(ϕi) $\leq$ time-of-last-access(ϕj). For each ϕi, there are three possibilities: ϕi is in the CC, ϕi is not in the CC but is in the system's belief space, or ϕi is in neither the CC nor the system's belief space. We these up in turn.

A. ϕi is in the CC

Either ϕi can be proved from CC – ϕi or it cannot. If it can, then it can be dropped without loss, since any inferences which can be made using it can also be made without it, and apparently they are. Before dropping ϕi, however, the system must find any propositions in the belief space in whose set-of-support ϕi occurs and replace it in those sets-of-support by the set-of-support of the proof of ϕi from CC – ϕi. (In SNeBR, finding these propositions is simply checking for a link, and does not require scanning the knowledge base. Implementations on other systems may involve a higher search cost.) The system can then forget ϕi (i.e. delete it from the knowledge base). This is the provision which makes the order in which stale propositions are considered matter. Suppose that ϕi and ϕj both belong to CC. Then it may be that ϕi can be inferred from CC – ϕi and that ϕj can be proved from CC – ϕj, but ϕi cannot be proved from CC – $\{\phi i,\phi j\}$ and neither can ϕj. In this case, only one can be forgotten. Since the one considered first will be the one forgotten, the one that has gone unused longer should be considered first.

In the second and more interesting case, ϕi can not be derived from CC – ϕi. Either ϕi occurs in one or more sets-of-support, or it does not. If it doesn't, then it seems reasonable to forget it: if the system has never needed this piece of information before, it is unlikely to do so later. (Hypotheses which are anticipated to be needed rarely but crucial when they are can be flagged to prevent deletion.) A slightly more liberal strategy would drop ϕi provided that it occurs only in the sets-of-support of stale propositions; this effect can be obtained by ordering ϕ so that all members of the system's belief space that are not in the CC come before any that are.

If ϕi can not be proved from CC – ϕi but does occur in sets-of-support, it may still be inferrable from the belief space. Since everything in the belief space that is not in CC can be inferred from it, the only beliefs not in CC that we have to consider are those with ϕi in their set-of-support. Let $B(\phi i) = \{\psi \mid \psi$ is in the system's belief space, $\psi \notin \phi$, and ϕi is in the set-of-support of $\psi\}$, and consider $CC' = CC \cup B(\phi i) - \phi i$. Suppose that ϕi can be proved from CC'. This means that CC' contains the information of ϕi, in a form that the system finds more useful: it is actually accessing the members of $B(\phi i)$, but not ϕi itself.

Suppose there is only one proof of ϕi from CC'. Let $B'(\phi i)$ be the subset of $B(\phi i)$ appearing in the set-of-support of that proof. Then add $B'(\phi i)$ to CC, replace ϕi by $B'(\phi i)$ in all sets-of-support, and forget ϕi. This restructures the CC to adapt to the actual pattern of use.

Suppose on the other hand that there are several proofs of ϕi from CC' and that they have different associated sets $B'(\phi i)$. Then the system must choose which propositions in its belief space to elevate to the CC. The system could choose the smallest set. This corresponds to the principle that axiom sets should be kept as small as possible, and so favors space over time. Alternatively, it could choose the set whose elements collectively have the largest number of recent references. This is the more interesting option, because the more adaptive: it adds the propositions most used and hence presumably most useful. It is also possible to combine these, to select a relatively small set with relatively high usefulness.

B. ϕi is in the system's belief space but not in the CC

Before forgetting stale members of the system's belief space, the system checks whether they suggest useful generalizations. Suppose that ϕi contains at least one individual constant. Check the belief space for other propositions $\phi i'$ which differ from ϕi only in individual constants. (In the case of SNeBR, this check costs relatively little because the propositions in question share structure with ϕi at all relational constants. In other specific architectures, it may be more costly.) If other such $\phi i'$ exist, form the proposition fi* by replacing all individual constants that the $\phi i'$ do not hold in common by variables and then quantifying universally over those variables. If ϕi^* is already in the system's belief space, do nothing at this point. If it is not, try to deduce it, and if successful, add it to the belief space. If ϕi^* can not be deduced, try to deduce $p\phi i^*$.

Next, check ϕ for other propositions which have at least one individual constant in common with ϕi. For

each such ϕj, consider the pair $\{\phi i, \phi j\}$ and look for pairs $\{\phi i', \phi j'\}$ where $\phi i'$ corresponds to ϕi as above, $\phi j'$ corresponds to ϕj, and $\phi i'$ and $\phi j'$ share the corresponding constant. Let $\phi ij = \phi i \supset \phi j$, $\phi ji = \phi j \supset \phi i$, $\phi ipj = \phi i \supset p\phi j$, and $\phi jpi = \phi j \supset p\phi i$, and form ϕij^*, ϕji^*, ϕipj^* and ϕjpi^* as before. These represent hypotheses about "if-then" universal rules (and corresponding default generalizations) which might shorten inference paths. Try to deduce each of these from the CC; any which can be deduced should be added to the belief space. Finally, forget ϕi and any of the $\phi i'$ which are also in ϕ and not in the CC.

C. ϕi is in neither the CC nor the system's belief space.

In this case, ϕi is important to the system only if it is in a belief space which the systems "cares about". For ϕi to have entered the knowledge base without belonging to the system's belief space, one of several things must have been true. Either it was once in the belief space but left it because at least one of the hypotheses in the set-of-support for ϕi was dropped from the CC (not forgotten, but found to be false), or ϕi was deduced in the course of a deduction involving hypothetical reasoning, or ϕi was deduced in the course of reasoning about some other agent's beliefs.

If ϕi was involved in hypothetical reasoning, and if the context relative to which ϕi was deduced (or hypothesized) often matters to the system, the system may want to protect that context by flagging it in such a way that the rules for assimilating information in the system's belief space also hold relative to that context's belief space. This amounts to saying that some hypothetical situations matter enough to the system that it is worth maintaining information about them the same way that it is maintained about (what the system regards as) the actual situation. Likewise, if ϕi belongs to the context or belief space of another agent about whom the system frequently must reason, the context for that agent may also be protected. If ϕi is an hypothesis of an unprotected context, forgetting ϕi involves also forgetting propositions in whose set-of-support ϕi figures. The strategy therefore prohibits forgetting ϕi if it occurs in every set-of-support for one or more propositions not in ϕ, since the existence of such non-stale propositions indicates that the context, although unprotected, is still active. In all other cases, if ϕi does not belong to the belief space of any protected context (including the CC), it can simply be forgotten.

V. Discussion

A. Information Loss

Most of the strategy outlined above is straightforward both in implications and in implementation. Potentially controversial decisions surround instances of forgetting in which information is actually lost. These arise whenever something is forgotten which cannot be deduced from its context, that is, when hypotheses of active contexts are dropped. This happens when a stale hypothesis cannot be derived from the rest of the its context and has no "fresh" consequences.

In the case of the CC, the rationale for forgetting the hypothesis is that if it hasn't been needed yet, it probably won't be. How good this rationale is depends on how long the system has been around: in the early phases of system use, some areas may simply not have been got to yet. It follows that it may be reasonable to have a second latency time t_{cc} which is checked before a stale hypothesis in the CC is discarded. That is, to determine whether the hypothesis is stale, the same threshhold is used as for any other proposition. But before discarding the hypothesis, the system checks whether the hypothesis has been latent longer than t_{cc}. The more conservative the system, the longer t_{cc} should be. Alternatively, protection can be taken as absolute for hypotheses: they can be replaced by other propositions so long as they remain deducible, but in no case can they be forgotten without being reconstructible.

For hypotheses of unprotected contexts, the rationale for forgetting the information is that it is the only way to get rid of outdated contexts. Since hypothetical contexts arise routinely in the course of reasoning with certain inference rules, it is desirable to be able to get rid of them later: contexts and intermediate conclusions which existed only in order to work through a single proof-by-cases constitute true clutter, and should be forgotten. The price for this is that if for any reason a context which should be protected isn't, information about it may be irretrievably lost.

B. Extensions of the Strategy

The strategy can be extended in several interesting ways. The most intriguing possibility occurs when ϕi belongs to CC, and when ϕi can almost be inferred from CC', but not quite. That is, most of the information of ϕi has been captured by inferences already made, but there is some $\phi i\dagger$ which is not there. What we would like the system to do is find simultaneously the weakest and the most interesting $\phi i\dagger$ such that $CC' + \phi i\dagger$ entails ϕi. The system would then add $\phi i\dagger$ and the appropriate $B'(\phi i)$ to CC, making the correct alterations to sets-of-support and forgetting ϕi. The obvious problem here is to get the system to formulate $\phi i\dagger$. Other options include finding more sophisticated patterns in stale instances than the simple implications described above, and considering when the system should take information as suggesting (default) generalizations even when it can't prove them.

C. Computational Cost of the Strategy

Despite the features of SNePS and SNeBR mentioned above which reduce search costs, assimilation passes are obviously very expensive. Worse, their cost rises rapidly as the number of stale propositions increases, since the amount of inference required grows rapidly. This might seem to suggest that passes should be run frequently (at least relative to t), to hold down the size of ϕ. Unfortunately, the usefulness of each pass also increases in proportion to the size of ϕ. This suggests that passes should be infrequent, ideally off-line or at very low use times, and that t should be large. The

latter also follows from the desire not to discard potentially useful intermediate results too quickly. On the bright side, as the system adapts to its environment, the number of major changes should decrease, resulting in a natural reduction in assimilation cost.

VI. Conclusion

The strategy described here provides mechanisms for systems to assimilate information in response to the actual patterns in which they have been called on to use that information. It allows them to eliminate clutter while retaining useful intermediate deductive results, thus avoiding repeating inferences while lessening the cost in space. More interesting, it also lets them restructure their belief spaces, promoting important derived principles to axiom status and demoting less useful axioms to belief status or forgetting them altogether. This lessens the need for system designers to anticipate the environment in which the system will be used (including the precise questions it will be asked) by letting systems adapt their axiom structures to their use environments, and within a single environment to changes in emphasis over time. The strategy is indifferent to the area of application, and while it was designed for a particular knowledge base architecture, it can be readily adapted to other architectures so long as they retain an essentially axiomatic structure (that is, so long as they have a distinction between context and belief space). It is thus a very general approach to self-reorganization in the particular area of information assimilation.

References

[Martins, 1983a] João P. Martins. Belief revision in MBR. In *Proceedings of the 1983 Conference on Artificial Intelligence*, Rochester, Michigan, 1983.

[Martins, 1983b] João P. Martins. *Reasoning in Multiple Belief Spaces.* Ph.D. Dissertation, Technical Report 203, Department of Computer Science, SUNY at Buffalo, May 1983.

[Martins and Shapiro, 1986a] João P. Martins and Stuart C. Shapiro. Theoretical foundations for belief revision. In Joseph Y. Halpern, editor, *Theoretical Aspects of Reasoning About Knowledge*, pages 383–398, Morgan Kaufmann Publishers, Los Altos, California, 1986.

[Martins and Shapiro, 1986b] João P. Martins and Stuart C. Shapiro. Hypothetical reasoning. In *Applications of Artificial Intelligence to Engineering Problems: Proceedings of the First International Conference*, pages 1029–1042, Southampton, U.K., University of Southampton, April 1986.

[Martins and Shapiro, 1986c] João P. Martins and Stuart C. Shapiro. Belief revision in SNePS. In *Proceedings of the Sixth Canadian Conference on Artificial Intelligence*, pages 230–234, Montréal, Quebec, Canadian Society for Computational Studies of Intelligence, May 1986.

[Martins and Shapiro, 1984] João P. Martins and Stuart C. Shapiro. A model for belief revision. In *Non-monotonic Reasoning Workshop*, pages 241–294, New Paltz, New York, American Assocation for Artificial Intelligence, October 1984.

[Martins and Shapiro, 1983] João P. Martins and Stuart C. Shapiro. Reasoning in multiple belief spaces. In *Proceedings IJCAI-83*, pages 370–373, Karlsruhe, Federal Republic of German, International Joint Committee for Artificial Intelligence, August 1983.

[Nutter, 1983a] Jane Terry Nutter. Default reasoning using monotonic logic: a modest proposal. In *Proceedings AAAI-83*, pages 297–300, Washington, D.C., American Association for Artificial Intelligence, August 1983.

[Nutter, 1983b] Jane Terry Nutter. *Default Reasoning in A.I. Systems.* Technical Report 204, Department of Computer Science, SUNY at Buffalo, October 1983.

[Shapiro, 1979] Stuart C. Shapiro. The SNePS semantic network processing system. In Nicholas V. Findler, editor, *Associative Networks: The Representation and Use of Knowledge by Computers*, pages 179–203, Academic Press, New York, 1979.

[Shapiro and Rapaport, 1986] Stuart C. Shapiro and William J. Rapaport. SNePS considered as a fully intensional propositional semantic network. In *Proceedings AAAI-86*, pages 278–283, Philadelphia, Pennsylvania, American Association for Artificial Intelligence, August 1986.