Making SME greedy and pragmatic

Kenneth D. Forbus Dan Oblinger
Qualitative Reasoning Group

Beckman Institute, University of Illinois
Phone: (217) 333-0193; Internet: forbus@p.cs.uiuc.edu

Abstract: The Structure-Mapping Engine (SME) has successfully modeled several aspects of human analogical processing. However, it has two significant drawbacks: (1) SME constructs all structurally consistent interpretations of an analogy. While useful for theoretical explorations, this aspect of the algorithm is both psychologically implausible and computationally inefficient. (2) SME contains no mechanism for focusing on interpretations relevant to an analogizer's goals. This paper describes modifications to SME which overcome these flaws. We describe a *greedy merge algorithm* which efficiently computes an approximate "best" interpretation, and can generate alternate interpretations when necessary. We describe *pragmatic marking*, a technique which focuses the mapping to produce relevant, yet novel, inferences. We illustrate these techniques via example and evaluate their performance using empirical data and theoretical analysis.

1 Introduction

The importance of analogy in human reasoning makes it a natural focus for cognitive simulation. The Structure-Mapping Engine (SME)[6,7], has been used to successfully model several aspects of human analogical processing. As a simulation of Gentner's Structure-Mapping theory [9,10], SME has been used to model human soundness judgements [13], to study the representational and processing choices in analogical processing [8], and as part of a model of sequence learning [14]. SME has also been used in an AI system which learns qualitative physics by analogy [3].

We believe several features of SME are accurate reflections of human analogical processing, including the emergence of global interpretations from local matches, the use of structural evaluation criteria as a default means of judging a comparison, the ability to generate novel candidate inferences, and the ability to construct and compare multiple interpretations of a comparison. However, the current SME algorithm has several drawbacks. First, SME constructs all structurally consistent interpretations of an analogy. This is often useful for theoretical explorations, since it allows one to know for certain the best possible interpretation of a given comparison. But it is extremely implausible psychologically. There are in the worst case a factorial number of potential solutions, making exhaustive enumeration impossible under any reasonable assumptions about human processing constraints. Even for theoretical explorations, as we tackle more realistic representations (c.f., [8]) this aspect of SME has become a stumbling block. To use SME as a central component in larger-scale simulations and AI systems, a more practical algorithm is needed. Here we describe a *greedy algorithm* which efficiently provides good approximations to the "best" interpretation. Although any greedy algorithm must sometimes fail to deliver optimal solutions, we demonstrate that in fact on this task it performs superbly.

The second drawback is that SME does not focus the mapping process according to the goals of the system. Such influences can be incorporated into analogical processing in several ways. Standard structure-mapping postulates that goals help determine both what gets matched and how the match is evaluated, but excludes them from the mapping stage itself [11]. Holyoak and Thagard's ACME model [12] blends structural, semantic, and pragmatic considerations into weights in a connectionist network, using a relaxation scheme to derive a single solution as an approximate best mapping. In addition to biasing preference for correspondences according to relevance, they allow *queries* to be inserted in the target description. If the query is supported by the match it is construed as the candidate inference of the analogy. A different approach is used by Falkenhainer's *contextual structure mapping* [3,5], which provides an elegant account of how to relax both the identicality and 1:1 constraints of structure-mapping when doing so provides more useful conjectures for the analogizer.

This paper describes a new technique, *pragmatic marking*, which is consistent with both standard and contextual structure mapping. The idea is to filter what subsets of local matches are

Figure 1: An example of SME input descriptions

We use this simple example for illustration only, realistic representations are typically much larger. The italicized numbers are not part of the representation, but have been introduced to provide a convenient means for refering to subexpressions later on.

Base domain:

```
1 (IMPLIES  2 (AND  3(SENSITIVE-TO  4 LITMUS32  5 ALCOHOL-VAPOR)  6 (INSIDE  7 COOLANT  8 SUMP)
                                  10 (HELD-CLOSE LITMUS32 SUMP))
                    11 (DETECTABLE  12 (GIVES-OFF COOLANT ALCOHOL-VAPOR)))
13 (IMPLIES  14 (LIQUID COOLANT)  15 (POSSIBLE (GIVES-OFF COOLANT ALCOHOL-VAPOR)))
16 (IMPLIES  17 (DECREASED  19 (PRESSURE SUMP))
              20 (INCREASED  21 (FLOW-RATE  22 (FLOW  23 STILL SUMP COOLANT  24 PIPE))))
26 (IMPLIES  27 (INCREASED (PRESSURE SUMP))
                    28 (DECREASED (FLOW-RATE (FLOW STILL SUMP COOLANT PIPE))))
29 (IMPLIES  30 (DECREASED  31 (AREA PIPE)) (DECREASED (FLOW-RATE (FLOW STILL SUMP COOLANT PIPE))))
32 (IMPLIES  33 (INCREASED (AREA PIPE)) (INCREASED (FLOW-RATE (FLOW STILL SUMP COOLANT PIPE))))
33 (CAUSE  34 (GREATER  35 (PRESSURE STILL) (PRESSURE SUMP)) (FLOW STILL SUMP COOLANT PIPE))
36 (FLAT-TOP COOLANT)
```

Target domain:

```
1 (INCREASED  2 (FLOW-RATE  3 (FLOW  4 EFFLUENT  5 HEAT-SINK  6 HEAT  7 HX)))
8 (DETECTABLE  9 (GIVES-OFF EFFLUENT  10 RADIATION))
11 (CAUSE  12 (CONTAINS EFFLUENT  13 STRONGTIUM) (GIVES-OFF EFFLUENT RADIATION))
14 (LIQUID EFFLUENT)
15 (FLAT-TOP EFFLUENT)
16 (GREATER  17 (TEMPERATURE EFFLUENT)  18 (TEMPERATURE HEAT-SINK))
```

considered by whether or not they can support candidate inferences relevant to the analogizer's stated goal. Unlike ACME's query mechanism, this technique does not require the actual form of the candidate inference to be specified in advance. Thus our technique is better able to support the use of analogy in modeling problem-solving and discovery.

Section 2 reviews the SME algorithm using an example. Section 3 describes the *GreedyMerge* algorithm for efficiently combining local matches into consistent global interpretations. We analyze its theoretical properties and we demonstrate empirically that it tends to be optimal, in that the first interpretation it provides is usually the same as the best interpretation found by the exhaustive SME merge algorithm. Section 4 describes pragmatic marking, analyzes its complexity, and illustrates it by example. Finally, we discuss our plans for future work.

2 How SME works

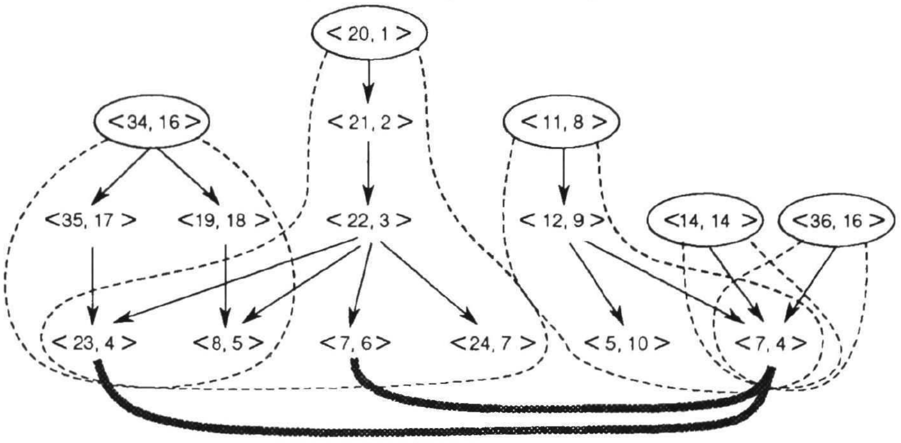
Here we sketch the standard SME algorithm to provide the backdrop for our improvements (see [7] for details). SME takes as input two propositional descriptions, a *base* and a *target*. It produces as output a set of global interpretations (gmaps) of the comparision. Each gmap contains a set of *correspondences* linking items in the base and target (including both entities and statements about them), a *structural evaluation score* which provides an indication of match quality, and a set of *candidate inferences*. The candidate inferences are statements in the base which can be hypothesized to hold in the target as a result of the gmap's correspondences. Each candidate inference is a surmise, and hence must be evaluated by other means to ensure its validity.

Figure 1 shows a simple example we use through the paper for clarity. Consider a case-based design system, which already had designed a still and was now working on a recycling plant. The base domain shows part of what it might retain about the still, and the target shows part of the description of the new design. This analogy can help solve two problems: how one might detect radiation in the effluent and how one might increase the rate of waste heat removal.

SME begins the mapping process by computing *match hypotheses* (*MH*'s), each representing

Figure 2: Hypothesized local matches for the comparison

Each match hypothesis has the form < *Base*,*Target* >, where *Base* and *Target* are expression numbers from Figure 1. The roots of the graph are circled, and the pmap defined by each root is indicated by dotted lines. The thick lines indicate nogoods. Only structurally consistent *MH*'s are shown for clarity.



a potential correspondence between an item of the base and an item of the target[1]. Figure 2 depicts the match hypotheses for our example. These local matches must be carefully filtered and combined to build structurally consistent interpretations. First, MH's involving items whose arguments cannot be placed in correspondence are eliminated from further consideration. In our example, the hypothesized match between these two statements

```
B: (CAUSE (GREATER (PRESSURE STILL) (PRESSURE SUMP)) (FLOW STILL SUMP COOLANT PIPE))
T: (CAUSE (CONTAINS EFFLUENT STRONGTIUM) (GIVES-OFF EFFLUENT RADIATION))
```

fails because neither of the corresponding arguments can match, while

```
B: (DETECTABLE (GIVES-OFF COOLANT ALCOHOL-VAPOR))
T: (DETECTABLE (GIVES-OFF EFFLUENT RADIATION))
```

is locally consistent, given the hypothesized pairings between COOLANT and EFFLUENT and between ALCOHOL-VAPOR and RADIATION. (These pairings can be considered as the *arguments* of the match hypothesis.) Next, SME installs local consistency constraints (*nogoods*) between pairs of MH's to mark potential violations of the 1:1 constraint. That is, the MH which maps COOLANT to EFFLUENT cannot ever be part of the same interpretation as the MH which maps COOLANT to HEAT. These local inconsistencies are propagated up the argument structure of the match hypotheses, to rule out MH's whose argument matches do not suggest consistent correspondences. Those MH's which remain become the grist for gmap construction.

Constructing maximal sets of MH's is the goal of gmap construction. A gmap is maximal if adding another MH causes structural inconsistancy. It is useful to view the set of match hypotheses as a partial order, with the MH's concerning object correspondences forming the bottom elements and inclusion relationships determined by the argument structure. Call an MH a *root* if it is consistent and is not an argument of some other match hypothesis. The roots of this graph are the initial gmap candidates, or pmaps, for "partial mappings" (Again, see Figure 2).

So far, the computational complexity is low. If n is the number of items in the base and target, then finding match hypotheses and local inconsistent combinations are both $\mathcal{O}(n^2)$, and the various propagation steps are $\mathcal{O}(log(n))$. Exhaustively combining pmaps into gmaps is the expensive part. It begins cheaply, by taking the union of the constraints for each pmap's correspondences to

[1] The rules which guide MH construction are programmable. To simulate structure-mapping, attributes and relational items must have identical functors. Different rule sets can be used to implement context-sensitive methods for relaxing identicality [5] and even simulate certain aspects of ACME[4].

compute what it is inconsistent with $(\mathcal{O}(n^2))$. The standard SME algorithm builds every possible complete gmap by making successive merges, subject to these consistency constraints, until no larger combinations can be built. If p is the number of pmaps, there are at worst $p!$ gmaps. This, of course, is expensive. Typical examples perform much better than this, due to the strong filtering effects of structural consistency. As [7] reports, on many complex examples SME takes only a few seconds of CPU time. However, we have found examples that can produce thousands of gmaps, and take days of CPU time to compute.

Finally, the structural evaluation and candidate inferences for each gmap are computed. These operations are of low complexity [7]. The structural evaluation score computation is irrelevant for this paper, see [7,8] for details. The only important feature is that the structural score of a gmap is the sum of its MH's scores, so it can easily be computed for pmaps and combined during merging.

Candidate inferences are computed by finding structure in the base which is consistent with a gmap's correspondences, but is not in fact included in them. Thinking now of the base domain as a graph, we are seeking structures which are roots (e.g., they are not themselves arguments of another item) and which have some, but not all, of their subitems mapped by the correspondences. Such items comprise potential new knowledge about the target, and are carried over by making the substitutions defined by the correspondences. Skolem functions are provided for base objects not mentioned in the correspondences. One candidate inference from a gmap resulting from our example comparison is:

```
(CAUSE (GREATER (TEMPERATURE EFFLUENT) (TEMPERATURE HEAT-SINK)) (FLOW EFFLUENT HEAT-SINK HEAT HX))
```

because the base structures *34* and *22* can map onto the target structures *16* and *3* respectively, while *33* in the base has no correspondence in the target (see Figures 1 and 2).

We believe the ability to generate structurally justifiable conjectures about the target is a central feature of analogy, responsible for its important role in creative problem solving and discovery (c.f. [2]). The rest of this paper shows how to achieve uniformly low complexity in gmap construction (at the cost of not always providing the optimum answer) and how to tune SME to produce novel candidate inferences relevant to the analogizer's goals.

3 Greed

The *greedy method* is a standard technique for combining a set of constrained, local solutions into a good global solution. The idea is that (a) finding a global solution can be modeled as deciding which local solutions to include and (b) some indication of "quality" exists for ordering local solutions[1]. Roughly, it works like this: Pick the best local solution. This rules out other choices, namely those which are inconsistent with the one picked. Throw away those which are inconsistent with your first choice. Now augment your solution with the best of the remaining local solutions. Again, this may rule out further choices, so one continues filtering and selecting until no more choices remain. The result is a single solution which is often, but not always, optimal.

The simplest version of $GreedyMerge$ casts gmap construction as a sequence of decisions about which pmaps should be combined. The ordering is provided by the pmap's structural evaluation score. Starting with the largest, each pmap is merged into the solution under construction, unless doing so would violate structural consistency. If a pmap is inconsistent with the solution, it is skipped. By starting with the largest we improve our chances of getting the best solution.

The attraction of greedy methods is low complexity. Their drawbacks are (1) the solution may not be optimal and (2) obtaining useful alternative interpretations can be difficult. Whether or not the first problem is significant for natural representations is an empirical question addressed below. The second problem is very important. We view the ability to generate multiple interpretations of an analogy as critical. Even with a firm goal in mind, there can still be several ways to interpret an analogy (c.f. the Contras example in [12]).

There are several ways that multiple interpretations could be generated. One algorithm we explored generated an approximation of the top n gmaps based on their structural evaluation. This is often not a good strategy. Consider a very large base and a medium-sized target, so that many small, semi-independent pmaps are formed as well as several large ones. The gmaps for such comparisons can often be divided into several families of basically different interpretations, with each family member varying only in which small pmaps are included. In such cases the top n gmaps

Figure 3: Greedy Merge Algorithm

We assume that the standard SME algorithm has been executed up to the stage of constructing pmaps.

```
1. Place pmaps in descending order based on their structural evaluation score.
2. PMAPS $\leftarrow$ the set of pmaps; USED $\leftarrow \{ \}$
3. Repeat for desired number of interpretations
        3.1. MAPPING $\leftarrow \{\mathrm{PMAP}_i\} \ni \mathrm{PMAP}_i \notin$ USED
             and $\forall j < i$   $\mathrm{PMAP}_j \in$ USED
        3.2. For each PMAP$\in$ PMAPS
             3.2.1. If PMAP is consistent with MAPPING Then
                    3.2.1.1. MAPPING $\leftarrow$ MAPPING $\cup$ PMAP
                    3.2.1.2. USED $\leftarrow$ USED $\cup$ {PMAP}
        3.3 Output MAPPING
```

Figure 4: Empirical Results of *GreedyMerge*

	Types Of Analogies		
	Object	Physical Systems	Stories
Number of matches	8	20	28
Min/Max number of gmaps	1/3	3/81	3/160
Min/Max GreedyMerge time (Sec.)	0/0.6	0.03/1.3	0.5/7
Min/Max FullMerge time (Sec.)	0/2.6	0.6/235	0.6/3335
Percentage of cases Greedy is optimal	100%	85%	96%
Lowest ratio Greedy Score / Best Score	100%	67%	91%

are likely to be trivial variations on the top theme, and since these will largely share the same set of candidate inferences, this is often undesirable. What we usually want is an alternate interpretation which is radically different. This suggests generating subsequent gmaps by starting with pmaps which are as large as possible but inconsistent with previously generated interpretations.

The algorithm we currently use (see Figure 3) starts by greedily generating the best gmap, and ensures that its gmaps are representative by always starting an alternate interpretation from a seed pmap which has never been used in any other interpretation. By adding the unused pmap first we ensure that we get a significantly different interpretation – it must be different since it contains a (hopefully large) structure which is inconsistent with all previously generated gmaps. Since the candidate inferences are based on the *MH*s these are also likely to be different. Note *GreedyMerge* reduces to the original greedy algorithm when generating only one interpretation. Each successive gmap starts with the largest unused pmap. All interpretations generated are maximal since *GreedyMerge* always attempts to add all pmaps to the interpretation during construction.

GreedyMerge is $\mathcal{O}(nlog(n))$ in the number of pmaps, and $\mathcal{O}(n)$ in the number of interpretations generated. The number of pmaps is $\mathcal{O}(n^2)$ in the size of the base and target, in worst case. In practice the number of pmaps tends to be much smaller, since only plausible *MH*'s are generated, and these tend to cluster into reasonably large pmaps.

GreedyMerge has been tested on over fifty different analogies, ranging from comparisons between physical phenomon, short stories, and object descriptions, drawn from the library of SME examples. Figure 4 summarizes the results. The stories show the most dramatic speedup – one story could not be included because the exhaustive algorithm failed to terminate after several days of computation, yet *GreedyMerge* found a reasonable interpretation in under a minute. And in most cases the first gmap generated by *GreedyMerge* was identical to the best gmap found by the exhaustive merge algorithm.

Why does *GreedyMerge* do so well? Typically, these large examples have a few large pmaps, only some of which are mutually inconsistent, and a much larger set of small pmaps. Thus the first few decisions are the really critical ones, and they are relatively easy to make. When will *GreedyMerge* fail? There are two kinds of cases where it should do poorly. The first is when there are many large pmaps with a high degree of mutual inconsistency, since many more decisions have to be correct, and hence the chance of error grows. This was the problem in the few cases (4 out of 56) where a non-optimal solution was generated. The second is when an initial, large pmap is inconsistent with every member of a large set of small but mutually compatible pmaps which in fact outweigh the initial one. We do not know how likely such situations are in natural representations. Fortunately, the ability to generate radically different interpretations provides a way to recover from such problems.

4 Pragmatism

The power of analogy comes from its ability to shed new light on the target by importing knowledge from the base. Retaining this ability using *GreedyMerge* requires modifying SME further. The reason is that the structurally best match may not always provide the most relevant inferences [3,5]. Returning to our example, the structurally best interpretation places the two flows in correspondence. But what if our goal is to propose how to detect strongtium in the recycler's effluent? As we find below, an interpretation which maps COOLANT to EFFLUENT is better for this purpose, even though a smaller structure is mapped. When using the original SME merge algorithm, one simply searches the interpretations to find a relevant inference. Since *GreedyMerge* is not exaustive, we must take care to ensure that relevant interpretations are actually generated.

Unfortunately, the techniques used by ACME provide no leverage here. Their techniques seem most useful for modeling instructional analogies, where a teacher may explicitly provide correspondences or point out which facts are most important. Here there is no correspondence involving the base fact that we wish to bring over, so it cannot be given extra weight or identified a priori as interesting. Introducing a query fact in the target does not help – if we knew the form of the query fact, we wouldn't need analogy to solve the problem. To get the novel inferences required for analogical problem solving requires a more generative solution.

Our ***pragmatic marking*** technique operates by looking for interpretations which can potentially import relevant base structure into the target. How can the relevant part of the base be found? Suppose we have target item G as our goal. That is, we want to find how G might legitimately be inferred on the basis of other (perhaps new) items in the target. For concreteness, suppose our goal is:

```
(DETECTABLE (GIVES-OFF EFFLUENT RADIATION))
```

Consider the set of match hypotheses generated for a comparison. The interpretations we are interested in must include a match hypothesis MH_G involving G, since only they can provide the structural grounds for candidate inferences involving G. (If G is not involved in any match hypotheses, then it cannot be the subject of any candidate inferences and hence we immediately know the comparison is useless for this purpose.) This means that the interpretation must in turn include MH's for the corresponding arguments of MH_G, and possibly for some larger structure of which it is a part.

Now consider the projection of MH_G onto the base domain. Again viewing the base as a graph, any pmap which includes the subgraph rooted in MH_G could provide inferences. However, pmaps which do not include this subgraph can also contribute to the structural grounding of an inference, so we must carefully examine them as well. There is some subset of roots of the base which contain MH_G's projection. Any pmap whose base projection lies outside this subset of the graph can be ignored, since it does not include the projection of our goal onto the base. Furthermore, any pmap inside this subset of the graph can be ignored if it is inconsistent with the correspondences implied by MH_G, since it could not be part of a gmap with it.

This intuitive picture provides the basis for the pragmatic marking algorithm (Figure 5). It is slightly more complicated to take into account the fact that there can be more than one MH_G, but otherwise is straightforward. The information required for the functions *TargetItem*, *BaseItem*, *Roots*, *BaseRoots*, *Descendants*, and *Nogood* is already computed in the process of generating pmaps. The complexity is thus $\mathcal{O}(|\{\text{pmap}\}| \times |\{MH_G\}|)$.

Figure 6 illustrates the results of two queries in our extended example. With the query about radiation detection, three out of the five pmaps are potentially relevant, and *GreedyMerge* successfully combines them all. The inference which results may be paraphrased as "By finding something which is sensitive to radiation, like litmus paper is sensitive to alcohol vapor, and holding it close to

Figure 5: Pragmatic Marking Algorithm

We assume that the standard SME algorithm has been executed, independent of any query, through pmap construction.

```
1. Let $\{MH_G\} = \{M | TargetItem(M) = G\}$
2. RELEVANT $\leftarrow \{\}$
3. For each $MH_G \in \{MH_G\}$,
      3.1 For each $\text{pmap}_i \in$ pmaps
            3.1.1 If $Descendants(\text{pmap}_i) \cap Descendants(MH_G) \neq \{\}$ then go to 3.1.4
            3.1.2 If $BaseRoots(\text{pmap}_i) \cap Roots(BaseItem(MH_G)) = \{\}$ then skip.
            3.1.3 If $Nogood(\text{pmap}_i, MH_G)$ then skip.
            3.1.4 Otherwise, RELEVANT $\leftarrow$ RELEVANT $\cup\ \text{pmap}_i$
4. $GreedyMerge$(RELEVANT)
```

Figure 6: Inferences generated in response to queries

G = (DETECTABLE (GIVES-OFF EFFLUENT RADIATION))

```
 There are 1 relevant interpretations:
GM1: 4 correspondences, SES = 2.5
  Object mappings: COOLANT <-> EFFLUENT, ALCOHOL-VAPOR <-> RADIATION
  Candidate Inferences:
(IMPLIES (AND (SENSITIVE-TO (:SKOLEM LITMUS32) RADIATION) (INSIDE EFFLUENT (:SKOLEM SUMP))
              (HELD-CLOSE (:SKOLEM LITMUS32) (:SKOLEM SUMP)))
         (DETECTABLE (GIVES-OFF EFFLUENT RADIATION)))
```

G = (INCREASED (FLOW-RATE (FLOW EFFLUENT HEAT-SINK HEAT HX)))

```
 There are 1 relevant interpretations:
GM5: 10 correspondences, SES = 4.375
  Object mappings: PIPE <-> HX, COOLANT <-> HEAT, STILL <-> EFFLUENT, SUMP <-> HEAT-SINK
  Candidate Inferences:
(CAUSE (GREATER (TEMPERATURE EFFLUENT) (TEMPERATURE HEAT-SINK)) (FLOW EFFLUENT HEAT-SINK HEAT HX))
(IMPLIES (INCREASED (AREA HX)) (INCREASED (FLOW-RATE (FLOW EFFLUENT HEAT-SINK HEAT HX))))
(IMPLIES (DECREASED (TEMPERATURE HEAT-SINK)) (INCREASED (FLOW-RATE (FLOW EFFLUENT HEAT-SINK HEAT HX))))
```

the effluent's container, one may detect when the effluent is giving off radiation." Notice that this interpretation is not the structurally best, which makes the flows correspond but is inconsistent with the mapping of COOLANT to EFFLUENT. The second question exploits the structurally larger interpretation, suggesting that in order to bring about an increase in the rate of heat removal, one can either increase the area of the heat exchanger HX or decrease the temperature of the heat sink. We have also successfully tested pragmatic marking on a variety of standard SME examples, with correct results in each case.

5 Discussion

We have seen how the SME algorithm can be modified to efficiently generate interpretations of analogies by using a greedy merging algorithm, and demonstrated that pragmatic marking can focus its efforts on just those interpretations likely to lead to relevant, novel candidate inferences. In moving from an exhaustive algorithm to a polynomial one we give up the guarentee of optimality, but as our empirical results indicate, we lose little by doing so.

There are several directions to explore next. For example, sometimes degrees of certainty or relevance can be estimated for items in a representation. It would be useful to exploit such information, as ACME does. Combining scores for certainty and relevance with the structural evaluation score used by *GreedyMerge* could provide an increased sensitivity to relevance that might be useful on larger problems. We also plan to use these techniques to embed SME into a larger simulation of human problem-solving activity.

6 Acknowledgements

This paper benefited from discussions with Dedre Gentner, Brian Falkenhainer, and Lenny Pitt. Gordon Skorstad helped draw figures, and Janice Skorstad provided valuable assistance with the SME data. This research was supported by the Office of Naval Research, Contract No. N00014-85-K-0559, an NSF Presidential Young Investigator award, and an equipment grant from IBM.

References

[1] Aho, A., Hopcroft, J. and Ullman, J. *Data Structures and Algorithms*, Addison-Wesley, Reading, Massachusetts, 1985.

[2] Burstein, M. H. Incremental learning from multiple analogies. In A. Preiditis (Ed.), *Analogica: Proceedings of the First Workshop on Analogical Reasoning*, London, Pitman Publishing Co., 1988

[3] Falkenhainer, B., An examination of the third stage in the analogy process: Verification-Based Analogical Learning, Technical Report UIUCDCS-R-86-1302, Department of Computer Science, University of Illinois, October, 1986. A summary appears in *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August, 1987.

[4] Falkenhainer, B., The SME user's manual, Technical Report UIUCDCS-R-88-1421, Department of Computer Science, University of Illinois, 1988.

[5] Falkenhainer, B. Contextual Structure Mapping. Unpublished manuscript.

[6] Falkenhainer, B., K.D. Forbus, D. Gentner, The Structure-Mapping Engine, *Proceedings of the Fifth National Conference on Artificial Intelligence*, August, 1986.

[7] Falkenhainer, B., Forbus, K., Gentner, D. The Structure-Mapping Engine: Algorithm and examples *Artificial Intelligence*, **41**, pp 1-63, 1989.

[8] Forbus, K. and Gentner, D. Structural evaluation of analogies: What counts? *Proceedings of the Cognitive Science Society*, August, 1989.

[9] Gentner, D., The structure of analogical models in science, BBN Tech. Report No. 4451, Cambridge, MA., Bolt Beranek and Newman Inc., 1980.

[10] Gentner, D., Structure-mapping: A theoretical framework for analogy, *Cognitive Science* **7**(2), 1983.

[11] Gentner, D., Mechanisms of analogical learning. To appear in S. Vosniadou and A. Ortony, (Eds.), *Similarity and analogical reasoning.* Presented in June, 1986.

[12] Holyoak, K. & Thagard, P. Analogical mapping by constraint satisfaction, *Cognitive Science* **13**, 295-355, 1989.

[13] Skorstad, J., Falkenhainer, B., Gentner, D., Analogical Processing: A simulation and empirical corroboration, in: *Proceedings of the Sixth National Conference on Artificial Intelligence*, Seattle, WA, August, 1987.

[14] D. Abstraction processes during concept learing: A structural view. In *Proceedings of the Tenth Annual Meeting of the Cognitive Science Society*, Montreal, August, 1988.