Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2006-071 November 7, 2006

Analogical Retrieval via Intermediate Features: The Goldilocks Hypothesis

Mark Alan Finlayson and Patrick Henry Winston

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

CSAIL

Analogical Retrieval via Intermediate Features: The Goldilocks Hypothesis

Mark A. Finlayson & Patrick H. Winston

{markaf, phw}@mit.edu

Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology,
32 Vassar St., Cambridge, MA 02139 USA

Abstract

Analogical reasoning has been implicated in many important cognitive processes, such as learning, categorization, planning, and understanding natural language. Therefore, to obtain a full understanding of these processes, we must come to a better understanding of how people reason by analogy. Analogical reasoning is thought to occur in at least three stages: retrieval of a source description from memory upon presentation of a target description, mapping of the source description to the target description, and transfer of relationships from source description to target description. Here we examine the first stage, the retrieval of relevant sources from long-term memory for their use in analogical reasoning. Specifically we ask: what can people retrieve from long-term memory, and how do they do it?

Psychological experiments show that subjects display two sorts of retrieval patterns when reasoning by analogy: a novice pattern and an expert pattern. Novice-like subjects are more likely to recall superficially-similar descriptions that are not helpful for reasoning by analogy. Conversely, expert-like subjects are more likely to recall structurally-related descriptions that are useful for further analogical reasoning. Previous computational models of the retrieval stage have only attempted to model novice-like retrieval. We introduce a computational model that can demonstrate both novice-like and expert-like retrieval with the same mechanism. The parameter of the model that is varied to produce these two types of retrieval is the average size of the features used to identify matches in memory. We find that, in agreement with an intuition from the work of Ullman and co-workers regarding the use of features in visual classification (Ullman, Vidal-Naquet, & Sali, 2002), that features of an intermediate size are most useful for analogical retrieval.

We conducted two computational experiments on our own dataset of fourteen formally described stories, which showed that our model gives the strongest analogical retrieval, and is most expert-like, when it uses features that are on average of intermediate size. We conducted a third computational experiment on the Karla the Hawk dataset which showed a modest effect consistent with our predictions. Because our model and Ullman's work both rely on intermediate-sized features to perform recognition-like tasks, we take both as supporting what we call the *Goldilocks hypothesis*: that on the average those features that are maximally useful for recognition are neither too small nor too large, neither too simple nor too complex, but rather are in the middle, of intermediate size and complexity.

Keywords: Analogy; analogical access; precedent retrieval; intermediate features; symbolic computational modelling.

Contents

1 Introduction & Background

We are concerned here with the retrieval of relevant precedents from long-term memory during an analogical reasoning task. That is we ask: when stimulated by a description, what descriptions do people retrieve from long-term memory, and how do they do it?

Analogical reasoning is generally split into three stages: retrieving, mapping, and transfer (French, 2002; Hall, 1989). The Structure-Mapping Theory by Gentner and coworkers (Gentner, 1983; Falkenhainer, Forbus, & Gentner, 1986) frames our understanding of the mapping stage. The theory states that shared structure is necessary to make an analogy between two descriptions. But, despite the importance of structure for making an analogy, psychological studies of the analogical retrieval stage indicate that it is extremely difficult for most people to recall structurally-related descriptions that would be useful in analogical mapping (Gick & Holyoak, 1980; Rattermann & Gentner, 1987); rather, people primarily retrieve descriptions that share superficial features, such as object or actor identity. Experiments have repeatedly shown the predominance of this so-called "mere-appearance" retrieval over analogical or structural retrieval, and models to date have attempted to account for the mere-appearance retrieval effect computationally.

Here, we note an important counterpoint to the predominance of mere-appearance retrieval, namely, that certain sorts of people can, in fact, consistently achieve structural recall (Chi, Feltovich, & Glaser, 1981; Schoenfeld & Herrmann, 1982; Shneiderman, 1977). These people often fall under the heading of "expert" for the domain in question. Previous models of retrieval that were tailored to understand the mere-appearance retrieval effect, do not account for the drastic differences between novice-like and expert-like retrieval (Thagard, Holyoak, Nelson, & Gochfeld, 1990; Forbus, Gentner, & Law, 1994). As a step toward addressing this gap, we present three computational experiments with two formally represented databases. The experiments suggest a single retrieval mechanism can account for both types of retrieval. Our model of the retrieval process works by splitting symbolic descriptions of situations into subgraphs (fragments) of all sizes and searching for approximate matches to these fragments in descriptions held in long-term memory. We show that, when features of a small size are used to perform retrieval, the model performs much like a novice. When the model uses intermediate features to effect retrieval, it approximates the performance of a expert.

Because our experiments suggest that fragments of an intermediate size are most useful for the purposes of retrieval, we say that these results support the *Goldilocks hypothesis*: the right fragment size for recognition-like tasks is intermediate, neither too large nor too small.

The layout of the paper is as follows. First we treat the psychological data that motivates the work (§1.1), and review Ullman's work on intermediate features which serves as inspiration for this model (§1.2). Then we give an overview of the retrieval model (§2.1), the experimental methodology (§2.3), and our dataset (§2.2). For further details of the implementation of the model, refer to the Appendix. Following a description of the Experiments (§3), we discuss the results (§4) and enumerate the contributions of the work (§5).

1.1 Psychological Data on Retrieval

Those who have studied analogy have been interested in how the retrieval process helps or hinders the discovery, construction, or application of an analogy. The general form of their questions has been: *Stimulated by some written or oral description, what do people retrieve from long-term memory?* The first specific instantiation of this question was: *Are subjects able to retrieve items that are profitable for analogical reasoning?*

The pattern of experiments used to attack this question has been straightforward. The experimental paradigm has been referred to as the "And-now-for-something-completely-different approach" (A. L. Brown, 1989); in it, the subjects are given some set of source tasks (such as memorizing a set of stories, or solving a problem), and then, after some delay, are given a second set of target tasks which are apparently unrelated to the source tasks. Nevertheless, unbeknownst to the subject, the two sets of tasks *are* related; the target tasks can be solved by relating them in an analogical fashion to the source tasks. Experimenters then vary the conditions of the experiment to determine the effect of variables of interest on the ability of subjects to retrieve and apply the source tasks to the target tasks.

Studies of this sort have provided strong evidence that most people cannot retrieve analogically profitable items from memory, even when delays between source and target tasks are short, or when analogical relationships are especially obvious. The seminal demonstration of this was by Gick and Holyoak (1980, 1983). In their experiments, subjects were asked to study a description of a military problem and its solution. Later the subjects were given a medical problem and asked to provide the solution. The medical problem was called the *radiation* problem because it dealt with using x-ray radiation to kill tumors, and was strongly analogous to the military problem that they had studied earlier. The experimental group was given a hint to consider earlier parts of the session for ideas, while the control group was given no hint. In the control group (no hint condition) only 30% of the subjects spontaneously noted and applied the analogy; remarkably, the experimental group (hint condition) had a solution rate of 75%. This suggests that over two-thirds of the subjects[1] were unable to spontaneously retrieve the analogous military problem on presentation of the radiation problem, even when the two problems were presented *within the same experimental session.*

Since Gick and Holyoak's study, many other experiments have provided evidence that people are unable to discover analogies without guidance. Gentner and Landers (1985) conducted a story recall experiment from which they concluded that people retrieve on the basis of surface similarity, rather than retrieving on the basis of analogical relatedness. Rattermann and Gentner (1987) went on to show that that object-descriptions and first-order relations between objects promote retrieval, but higher-order relations do not, and that ratings of inferential soundness and preference of retrieval are negatively correlated. Other problem solving and retrieval studies showed that subjects needed to be explicitly informed of the relation between two problems before they were able to apply analogical inferences, and that recall is heavily dependent on surface semantic or syntactic similarities between representations (Reed, Ernst, & Banerji, 1974; Reed, Dempster, & Ettinger, 1985; Ross, 1984, 1987). The pattern of retrieval shown by these experiments is clear: analogically related items are not preferred in retrieval.

Nevertheless, analogy is a process we draw upon every day in the cognitive tasks we perform. Indeed, the process of analogy is common enough to stimulate many cognitive scientists to study it. How do we reconcile this with the idea that analogically useful items are rarely retrieved? With this question in mind, some have investigated whether, under certain conditions, retrieval of analogically related items is preferred.

Holyoak and Koh investigated whether analogical retrieval was preferred if subjects were more experienced in the domain in question (Holyoak & Koh, 1987). Their experiment used psychology students who had been introduced in a classroom setting to the radiation problem given in Gick and Holyoak's experiments. These students were asked to solve a problem analogous to the radiation problem, the *lightbulb* problem. In Gick and Holyoak's work, only 30% of people were able to solve the radiation problem when they were exposed to the analogous military problem but not reminded of it; in Holyoak's experiment, 80% of the students solved the lightbulb problem without prompting or hints, indicating that a larger fraction had done structural encoding and retrieval. Along the same lines, Fairies and Reiser (1988) argued that if subjects were told they would be solving problems with the information they were provided, they would be more likely to encode structural features which would be useful for recall. They showed that in a problem solving environment subjects can encode structural features which can be useful later for structural recall. In studies of expert computer programmers, Shneiderman (1977) showed that expert computer programmers recall computer programs primarily based on the purpose of the code, but not on its specific form. Finally, working directly on analogical problem solving, Novick and coworkers provided convincing evidence that experts are better than novices at analogical retrieval (Novick, 1988; Novick & Holyoak, 1991). Novick's results were that, overall, subjects of greater expertise demonstrated greater positive analogical transfer and less negative analogical transfer—a finding that is consistent with the idea that expertise allows for reliable recall of analogs.[2]

[1] In a previous control experiment it was determined that radiation problem had a 10% base rate of solution; this indicates that, for the control and experimental groups respectively, at least 20% and 65% solved the problem by analogy, leading to the estimate of approximately two-thirds increase for the experimental group relative to the control.

[2] There was some doubt about this effect, in that the second experiment reported in that paper failed to expose an expert/novice difference in negative transfer. This discrepancy was explained, however, by reasoning that the more expert subjects postponed deep processing of the problems until they had realized that a quick, superficial retrieval based on surface features would not suffice. This was confirmed in the third experiment by providing experts with both a remote analog and a

Thus the psychological evidence is clear: there are two patterns of retrieval in the experimental literature relevant to analogical reasoning. In novice-like retrieval subjects do not naturally recall analogically-related items; to perform analogical retrieval they must be prompted and led to construct appropriate representations. In expert-like retrieval, subjects are able to recall structurally-related information in their domain of expertise, even if they may not do so immediately and spontaneously.

1.2 Intermediate Features in Visual Classification

Because our inspiration comes from work published by Ullman, Vidal-Naquet, and Sali (2002), we will now review those results to set the stage for our own work.

Ullman and coworkers noted that the human visual system assembles complex representations of images from simpler features, and then presumably uses those features later for storing, retrieving, and identifying related images. The question that naturally follows this is "which features are most helpful for these tasks, especially identification?"

They investigated this question by directly measuring the mutual information of various image fragments for specific visual classes. Mutual information is a formal measure of how good an indicator that feature is for presence of the class; it is the amount by which your uncertainty of the presence of the class is reduced when you know the feature is present. For their experiments, features were approximated by blocks of pixels (fragments) of images. They searched a database of approximately 180 faces and automobiles (that is, two classes) for fragments that matched well with a collection of approximately 50 selected face fragments of a wide variety of sizes. The complexity was varied by adding a variable amount of blur. Their measurements revealed that features of an intermediate size and complexity have, on average, the highest mutual information. For example, if you find what looks like a pair of eyes (an intermediate feature), you are more likely to have found a face than if you find what looks like a single eye (a small feature). Very large features, such as a whole face, are rarely found anywhere except in the original image, so they have low mutual information as well.

Using this information they designed a detection scheme that weighted intermediate features more heavily than either small or large features. This scheme produced a 97% face detection rate in novel images, with only a 2.1% false positives rate. They intuitively account for their detection scheme's impressive results by noting that small features produce many false positives (an eye often matches a random image feature by chance) and that large, complex features produce many false negatives (a detailed image of a face rarely matches anything in a collection of stored faces). Rather, it is the features of an intermediate size and complexity that are most useful.[3]

2 Overview

2.1 Retrieval Model

The psychological data reviewed in Section 1.1 shows that subjects display two sorts of retrieval patterns: a novice pattern and an expert pattern. Our underlying strategy to modelling this retrieval process is cast in a symbolic framework. In such a framework, written or oral stimuli that could be used with human subjects are translated into formal descriptions (graphs with labelled nodes) that capture salient semantic content. Inspired by Ullman's work, we propose that retrieval can be modelled by breaking the formal descriptions into pieces (subgraphs) and using these pieces as retrieval cues. We hypothesize that important difference between novice and expert subjects is the *informativeness* of the features that they use to retrieve

distractor source problem, providing the experts with a contrast and inducing them to more often recall and use the analog. Such an explanation is also supported by the results presented by Blessing & Ross (1996), where experts were found to rely to a certain extent on the superficial features of problems to predict deep structure (and therefore the solution procedure) when the two are correlated in the problem domain.

[3] Others in object recognition have also shown this same effect (Serre, Riesenhuber, Louie, & Poggio, 2002) and have applied the ideas of intermediate features to modelling object recognition in the brain (Lowe, 2000), and image indexing and retrieval for large databases (Obeid, Jedynak, & Baoudi, 2001).

precedents: novices, because of their inexperience, use relatively uninformative features, while experts use relatively informative features. Here, when we say *informativeness*, we mean this in an information-theoretic sense. Furthermore, we hypothesize that features of an intermediate size will be most informative by following the intuition used in Ullman's work, namely, that small features produce many false positives while large, complex features produce many false negatives, and so it it is those features that are not too small and not too large that are best. To test this hypothesis we constructed a retrieval model that can be tuned to focus on features of particular sizes when searching for precedents in memory. The model works as follows.

First, its inputs are graph descriptions of both a stimulus scenario (the target) and a collection of scenarios that are held in memory (the sources). Examples of these full graphs can be seen in Figure 1 and 15. We have a fairly controlled method for producing these descriptions from English text; this method is covered in Section 2.2, and the representations are covered in detail in Section A.1.

Second, the model breaks up the symbolic description of the target into features (graph fragments, or subgraphs) and searches for best matches to those features within the source descriptions in memory. There are examples of features of a small and intermediate size in Figure 17; an example of a large feature would be the whole graph. The features are analogous to the face fragments in Ullman's work. Searching for the features consists of constructing a match score between each feature in the target and each feature in every source. The match score between two features increases with their semantic similarity, structural similarity, and their size. Two features will have a low match score if they are semantically dissimilar, structurally dissimilar, or radically different in size. Each feature from the target description is then assigned a most-similar feature from each source description in memory (most-similar, that is, on the basis of the match scores). There are examples of match score calculations in Section A.4

Third, the model adds the match scores for each source together to form a retrieval score for that source relative to the target in question. This score is the final output of the model, and the magnitude of these scores are taken to indicate which descriptions are more or less likely to be retrieved. The model has two thresholds that control the range into which a match score must fall before it is added to the total. These thresholds indirectly control the average size of the features that are participating in the retrieval. The reason for applying the thresholds to the match score, instead of directly to the feature size, is discussed in Section 4.1.

The model is intended to demonstrate how intermediate features can enable expert-like retrieval. It is not intended as a model of exactly how retrieval is carried out *in the head*. The model is intended to support a hypothesis that is, in Marr's sense of the word, at the *computational* level. We hypothesize that expert-like analogical retrieval takes advantage of highly informative features, and these features will be on average of an intermediate size. This hypothesis entails that any retrieval method that uses feature-based retrieval and is intended to consistently effect analogical retrieval must, at some point, perform the same computations that are performed by our model. Because of this, such a retrieval process will display a dependance on feature size and informativeness similar to that shown by our model. This point is discussed in more detail in Section 4.

2.2 Datasets

We used two datasets of formal descriptions in our experiments. The first was a dataset of our own construction used in Experiments 1 and 2, which we will call the *Conflict* dataset because it consisted of descriptions of inter- and intra-national wars. The second dataset was the Karla the Hawk dataset, which has been used in other cognitive modelling studies (Gentner, Rattermann, & Forbus, 1993). The datasets are similarly organized: they contain a set of target descriptions and a set of source descriptions. Each individual target description has a set of matched source descriptions that are constructed so as to have a specific relationship to the target. This is important, because knowing the relationship between sources and targets beforehand allows us to determine what sorts of targets the model is retrieving, and thus make a determination if the retrieval is more novice-like or expert-like.

The classes of story relationships we used are not unique to our work, but follow on those established in the analogy literature (Gentner, 1983; Gentner & Landers, 1985): they include literally-similarly-related, analogically-related and merely-apparently-related sources. Mere-appearance sources share superficial, surface-semantic similarities with their target. Novices more often recall these sort of sources, and so are included in the database. Analogical sources are related to their target in a structural manner, but do not share surface-semantic details. Experts more often recall these sorts of sources, and so they are also included. Literally-similar sources correspond to a nearly exact duplicate of their associated target, and serve essentially as controls. Both novices and experts more often recall these sources than either mere-appearance or analogical sources. The Conflict dataset also contains Less-analogical sources, that share partial structure with their targets. The utility of the less-analogically similar descriptions is discussed in detail in Section 4.1.

The Karla the Hawk dataset was constructed by hand by Gentner and coworkers. In contrast, we constructed the Conflict dataset using a semi-automatic procedure. First we manually paraphrased simple target descriptions in simple, controlled English[4], then used a parser of our own construction on those descriptions to automatically construct formal target descriptions. The parser gives uniform results because it always produces the same graph structure when the input conforms to a recognized syntactic and semantic pattern, such as "X [move verb] toward Y", or "X caused Y."[5] These formal descriptions (graphs) can then be used directly with our retrieval model.

We synthesized sources by systematic transformation of the nodes and threads in the target. As already noted, we made four sources for each target description: an analogically related source (AN), a less-analogically related source (LAN), a mere-appearance source (MA), and a literally-similar source (LS).

The descriptions were all within the domain of inter- and intra-national armed conflict, and, with two exceptions, drawn from *The Encyclopedia of Conflicts since World War II* (Ciment, 1999). The target scenarios paraphrased were the 1989 civil war in Afghanistan (Ciment, 1999, p. 237), the Vietnamese invasion of Cambodia (p. 368), the Chad-Libya war (p. 389), the Chinese invasion of Tibet (p. 415), the 1962 China-India border war (p. 436), the 1969 China-USSR border war (p. 436), the 1979 war between China and Vietnam (p. 446), the 1997 civil conflict in the Congo (p. 487), the Bay of Pigs invasion of Cuba (p. 1,961), the 1968 Soviet invasion of Czechoslovakia (p. 533), the first Persian Gulf war (p. 811), and the Nigerian civil war (p. 1,038). Two additional paraphrases, of the American Civil War and American Revolution, were written without reference to a specific text.

Shown in Figure 1 is our paraphrase of the three and a half page description of China's invasion of Vietnam in 1979 from *The Encyclopedia of Conflicts*. The paraphrase is in simple, controlled English that is parsable by our system. Each paraphrase was between 9 and 28 sentences long (average 16.3, standard deviation 5.4). This description, given to the parser, produces a symbolic description graph, shown in Figure 2. The specific graph representations that are used in forming this description are covered in detail in Section A.1. In graph representation form the paraphrases contained from 39 to 110 nodes (average 65.3, standard deviation 20.2). Detailed characterization of the descriptions can be found in the Section 3.3.1 of Experiment 3, where they are contrasted with the Karla the Hawk dataset.

To generate the related sources for each target, we had a program perform a fixed generation procedure that operated on the graph representations. For example, suppose we began with a target description "The boy ran to the rock and hid because the girl threatened him." A description literally similar to this target has both highly-similar objects, but also a structurally and semantically similar relational and causal structure. In our implementation we replaced each object with another object which is nearly exactly the same. Thus we might replace *boy* with *man* and *girl* with *woman* and *rock* with *boulder* to produce "The man ran to the boulder and hid because the woman threatened him."[6]

A description that is merely-apparently similar to the original target description would be one that has highly similar objects and simple relations, but a different higher order relational and causal structure. To obtain a merely-apparently similar source, we leave the objects unchanged, but scramble the higher-order structure. This means that we take higher-order pieces and randomly switch their subjects and objects. This might produce "The girl threatened the boy because he ran to the rock and hid."

[4] Controlled languages have been used for some time to easily specify formal descriptions. See (Nyberg & Mitamura, 1996) and subsequent proceedings for descriptions of this technique.

[5] Although this English-to-graph-representation parser has not yet been described in the literature, it is not far removed in concept from the parser reported in previous work from our laboratory (Winston, 1980, 1982).

[6] In our representation, described in Appendix A.1, this meant that we replaced one object with another that matches on all but the last thread element.

The KhmerRouge controlled Cambodia.
Cambodia attacked Vietnam because the KhmerRouge disliked Vietnam and The KhmerRouge controlled Cambodia.
Vietnam disliked the KhmerRouge because Cambodia attacked Vietnam.
Vietnam attacked Cambodia because Cambodia attacked Vietnam.
Vietnam's army was larger than Cambodia's army.
Vietnam defeated Cambodia because Vietnam's army was larger than Cambodia's army and Vietnam attacked Cambodia.
Vietnam ousted the KhmerRouge because Vietnam defeated Cambodia.
China invaded Vietnam because Vietnam ousted the KhmerRouge.
Vietnam didNotWant China to invade Vietnam.
Vietnam's army impeded China invading Vietnam because Vietnam didNotWant China to invade Vietnam and China invaded Vietnam.
China left Vietnam because Vietnam's army impeded China invading Vietnam.

Figure 1: Sample Simple English Situation Description: China's Invasion of Vietnam in 1979

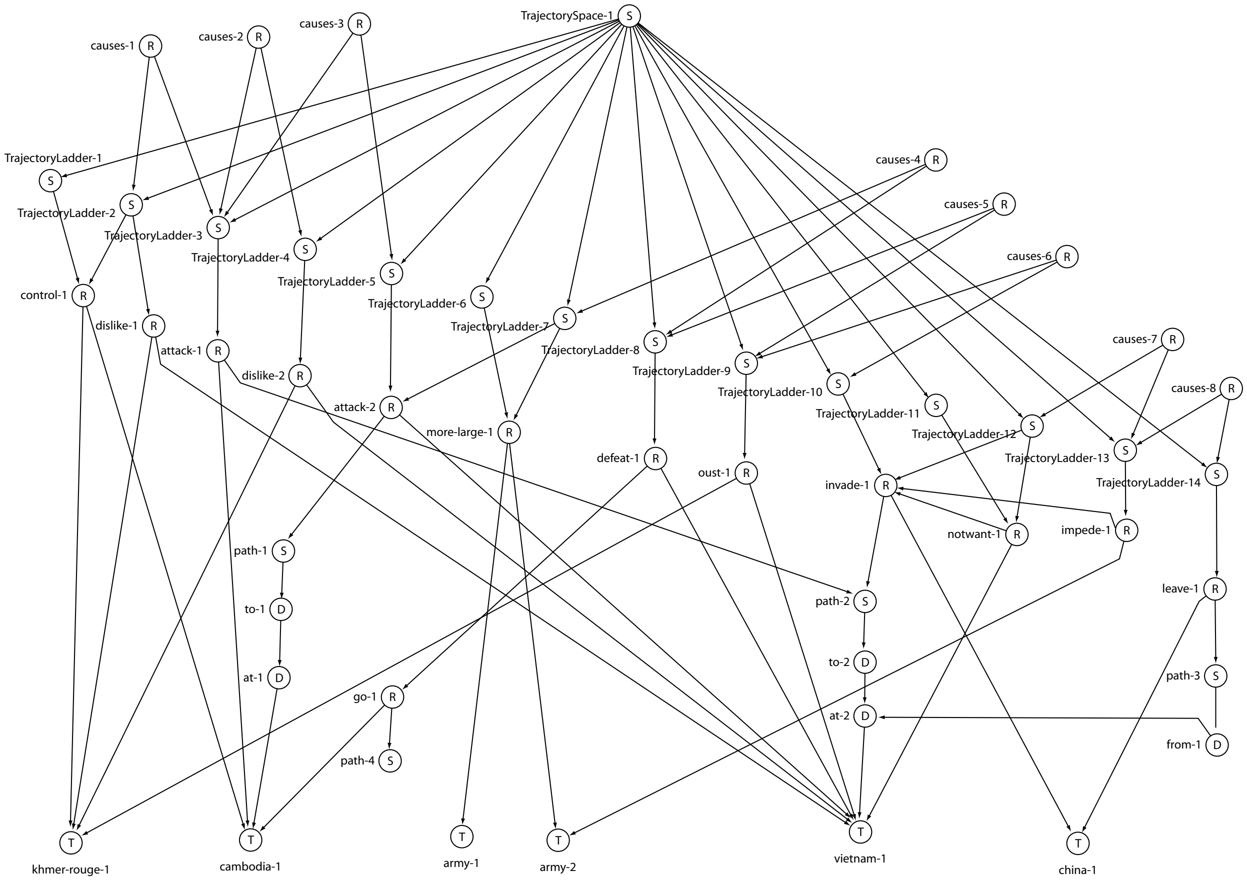


Figure 2: Sample Symbolic Description Graph: China's Invasion of Vietnam in 1979

A description that is analogically similar to the original target description is one in which higher-order structure is highly semantically similar, but the objects are not. To effect this we replaced all the objects in the target with objects which matched on only highest membership classes, while leaving the higher-order structure unchanged. Thus a generated analogical source might be "The army retreated to the castle and dug in because the enemy army approached."

A description that is less-analogically similar to the original target is one in which some fraction of the higher-order structure is highly semantically similar, but not all of it. To make one of these from the target, we transform as if to make an analogy, but we mix up the subjects and objects of some fraction of higher-order relations as is done for a mere-appearance match. For the experiments presented in this paper, the fraction was approximately one-third. This might produce "The army returned to their castle, but they only dug in when the enemy approached."

Table 1 summarizes the different sorts of source types and their examples.

Type	Example
Target	The boy ran to the rock and hid because the girl threatened him.
LS Source	The man ran to the boulder and hid because the woman threatened him.
MA Source	The girl threatened the boy because he ran to the rock and hid.
AN Source	The army retreated to the castle and dug in because the enemy army approached.
LAN Source	The army returned to their castle, but they only dug in when the enemy approached.

Table 1: Examples of systematic transformations of a target into literally similar (LS), merely-apparently similar (MA), analogical (AN), and less-analogical (LAN) descriptions.

2.3 Experimental Methodology

In our experiments, we adjusted the thresholds of the model to different values to control the average size of the feature participating in retrieval. As already noted, the hypothesis was that a small average feature size will achieve novice-like retrieval, and an intermediate feature size will achieve expert-like retrieval. In each experiment, the retrieval model was used to produce a retrieval score for each target relative to all the other targets and sources in the dataset at a range of thresholds, where each score indicates the strength of retrieval, and their order is the predicted retrieval order. The outcome was evaluated by looking at the relative strength of retrieval for all the different types of sources. For the novice-like pattern, the expected order of retrieval for sources related to the target was

$$\text{literally-similar} > \text{mere-appearance} > \text{analogical} > \text{less-analogical}$$

This matches the psychological data that novices should most often recall nearly exact matches (LS), but of remaining retrievals, prefer superficial matches (MA) more than structural matches (AN and LAN). For the expert-like pattern, we expected

$$\text{literally-similar} > \text{analogical} > \text{less-analogical} > \text{mere-appearance}$$

Again, this matches the psychological data in that experts, just like novices, prefer nearly exact matches (LS) over all others, but, unlike novices, prefer structural matches (AN and LAN) over superficial matches (MA). As can be seen, the difference between these two retrieval patterns is the position of the mere-appearance target. We formulated a simple measure of the goodness of fit of the order of retrieval scores to these patterns that is explained in detail in Section 3.1.2. The measure can be thought of as the answer to the following question: "Given the order of two sources, what is probability that those those sources are in the proper order?" The question can be asked for both novice-like and expert-like retrieval, thus producing two curves for each set of retrieval data.

3 Experiments

We performed three experiments to test the performance of our model. The first experiment demonstrated how the model moves from novice-like to expert-like retrieval as the average feature size is increased from small to intermediate. The second experiment showed that large features do not contribute significantly to the novice-like retrieval pattern. The third experiment reproduced the first experiment with the Karla the Hawk dataset. It showed only a modest effect consistent with our predictions. The results are discussed in detail in Section 4.1.

3.1 Experiment 1: Retrieval Simulation by Increasing Threshold

3.1.1 Procedure

This experiment used the Conflict dataset. As already mentioned, the Conflict dataset consists of fourteen target descriptions paraphrased in simple, controlled English and parsed into our graph representation, followed by the automatic generation of a set of four source descriptions from each target, for 56 sources in total. The retrieval model, described in algorithmic form in Section A.2, was used to measure the retrieval score between each target and all 56 sources for every setting of the lower threshold. The upper threshold was set at infinity. The algorithm is such that feature-feature matches with scores less than the lower threshold are not included in the final retrieval score, causing the average feature size to increase as the lower threshold is increased.

3.1.2 Results

The order of retrieval was compared to the predicted order for both novices and experts, resulting in two curves shown in Figure 3, graphed against the value of the lower threshold.

These curves show the probability of the model producing the correct human retrieval order calculated from the retrieval scores provided by the algorithm. As can be seen, the novice-like order is well predicted at a low threshold, that is, at a small feature size. The expert-like order is well predicted at intermediate feature sizes. The abscissa runs until the threshold is larger than the score of any feature-feature match score.

The probability curves were calculated as follows. Each source that was in the correct order relative to its associated mere-appearance target score was assigned a value of 1 (i.e., a correct prediction). If in the incorrect order, it was assigned a value of 0. If equal, they were given a value of 0.5 (indeterminate; fifty percent chance of choosing the correct order). These values were then averaged to obtain the probability of making a correct order prediction at a given threshold given the retrieval scores assigned by the model. As the novice-like or expert-like retrieval patterns differ only in the position of the mere-appearance source, only source positions relative to the mere-appearance source were considered.

The retrieval algorithm had in memory not only the sources generated from each target, but all other matches for all other targets as well; that is, there were 52 distractors in memory for every target. The retrieval algorithm always gave the four sources generated from each target the highest four scores, and all other sources—the distractors—came back with significantly lower scores.

For reference the raw values of the retrieval scores are shown in Figure 4. They are normalized at each threshold by dividing by the maximum possible score achievable at that threshold. This is the actual output of the retrieval algorithm, used to calculate Figure 3.

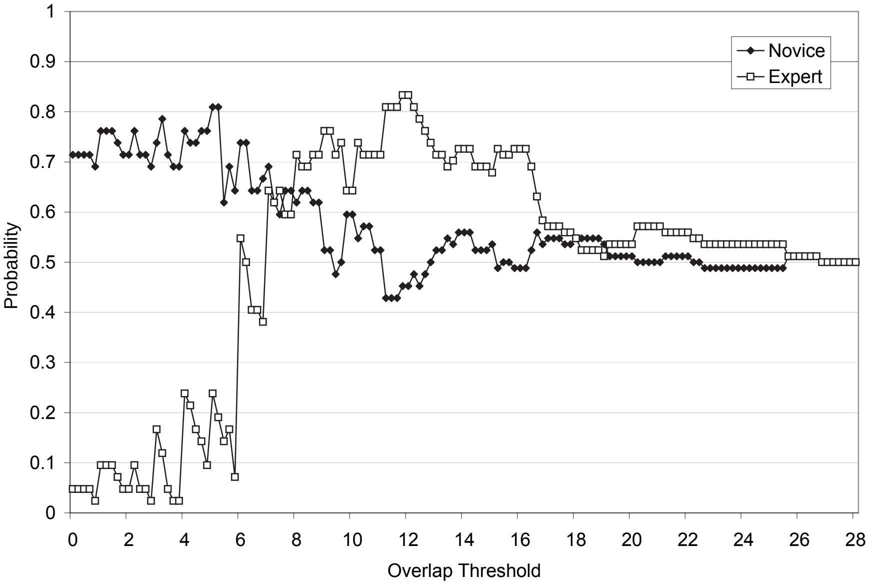


Figure 3: Probability of predicting the correct ordering (novice-like or expert-like) averaged over the dataset, against the lower threshold. The behavior of our test program begins with a novice-like behavior, becoming more expert-like when features of intermediate size begin to participate in matching, at a size threshold of about eight, but then the behavior degrades when only larger features are allowed to participate, at a size threshold of about 18.

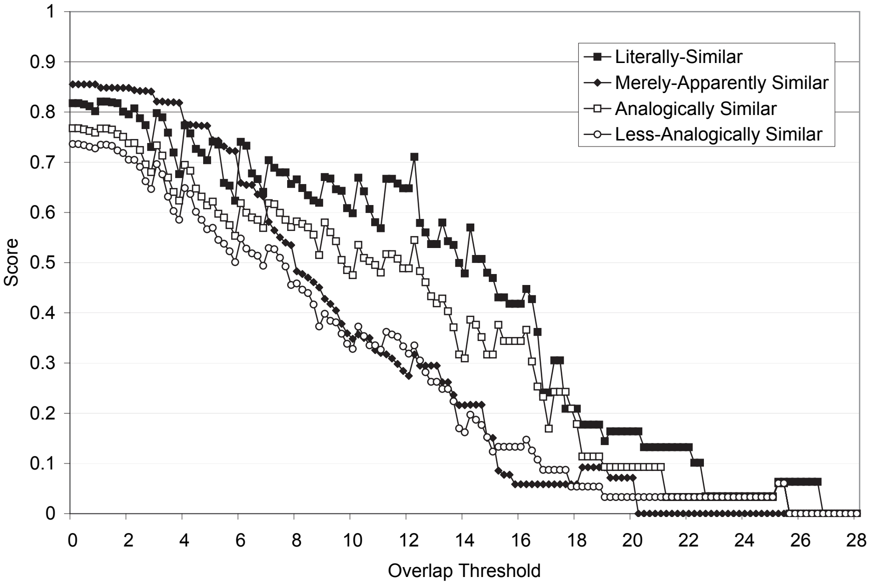


Figure 4: Raw retrieval scores as produced by the algorithm. Scores were normalized by dividing by the self-match score of the target description from which the sources are derived. As the threshold increases, fewer and fewer features are large enough to be counted, until no feature is large enough, and the scores all drop to zero.

3.2 Experiment 2: Retrieval Simulation by Decreasing Threshold

3.2.1 Procedure

This experiment used the Conflict dataset, and the same procedure as the first experiment, except that the lower threshold was fixed and the upper threshold was varied. This variation was performed to make sure that high-scoring feature matches are not contributing significantly to the novice-like retrieval pattern. High-scoring feature matches generally correspond to matches between large and intermediate, higher-order features, and these scores are included in the final score when the threshold is low (and also when it is intermediate). But, according to our hypothesis, novices do not use these features. To confirm that the inclusion of these high-scoring match pairs was not significant to the novice-like retrieval results in Experiment 1, we measured the retrieval behavior that resulted by discarding higher-scoring matches.

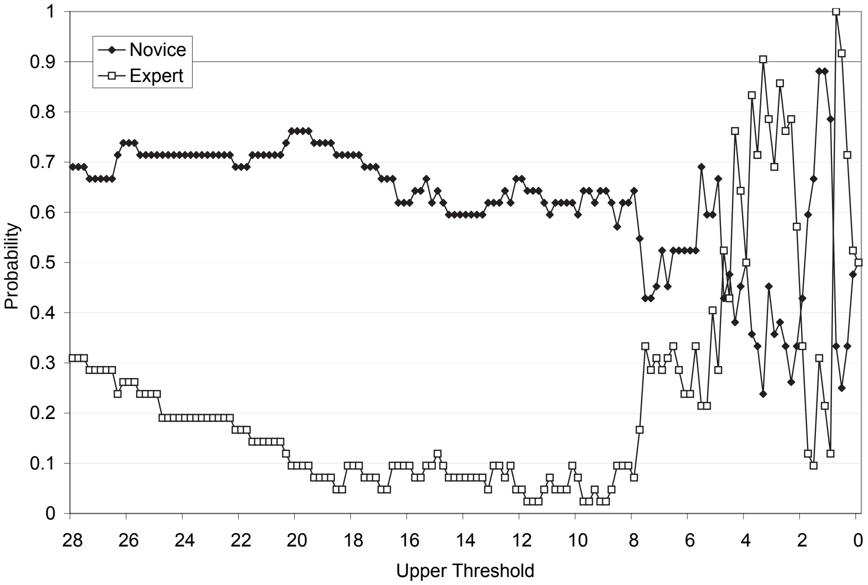


Figure 5: Probability of predicting the correct ordering averaged over the dataset, as the threshold is dropped from above. As expected, the behavior of our test program mirrors human novices until most of the features of small size no longer participate in matching, at which point the results fluctuate widely because relatively few features participate in the match.

For this experiment, we set the lower threshold at zero, and ran the model for all values of the upper threshold. When the upper threshold is higher than the largest feature score all features are included in the final retrieval score. When the upper threshold is zero no features are included, and all targets receive a score of zero. Because matches with scores larger than the upper threshold are discarded, the average feature size participating in retrieval decreases as the upper threshold decreases.

3.2.2 Results

Results for Experiment 2 are shown in Figure 5. As the upper threshold is lowered, the novice retrieval pattern is maintained until features with small scores begin to be discarded, at which point it begins to degrade. This result confirms that higher-scoring features do not significantly contribute to the novice retrieval pattern. Furthermore, because intermediate and small features together do not produce expert-like retrieval, and neither large-features (Experiment 1) nor small features (this experiment) alone produce expert-like retrieval, we conclude that intermediate features, or intermediate-features in conjunction with large features, are responsible for expert-like retrieval.

3.3 Experiment 3: Retrieval Simulation with Karla the Hawk Dataset

3.3.1 Procedure

For this experiment we used the Karla the Hawk dataset, which has been used with both human subjects and computational models (Forbus et al., 1994). We translated this dataset into our own representation, described in Section A.1. The Karla the Hawk dataset contained nine story sets, of which we used five[7]. We chose this subset of the Karla the Hawk stories because this subset was closest to our dataset in terms of the statistics of number of nodes and semantic tags (as measured by the average and variance). Each story set contains a target with related literally-similar, analogical, mere-appearance, and falsely-analogical source. A histogram of the average number of nodes broken down by node type, for both the Conflict dataset and the Karla the Hawk dataset, is shown in Figure 7, and a histogram of the average number of thread elements (semantic descriptions) is shown in Figure 8. Except for the dataset generation, which was omitted, the procedure was the same as in Experiment 1.

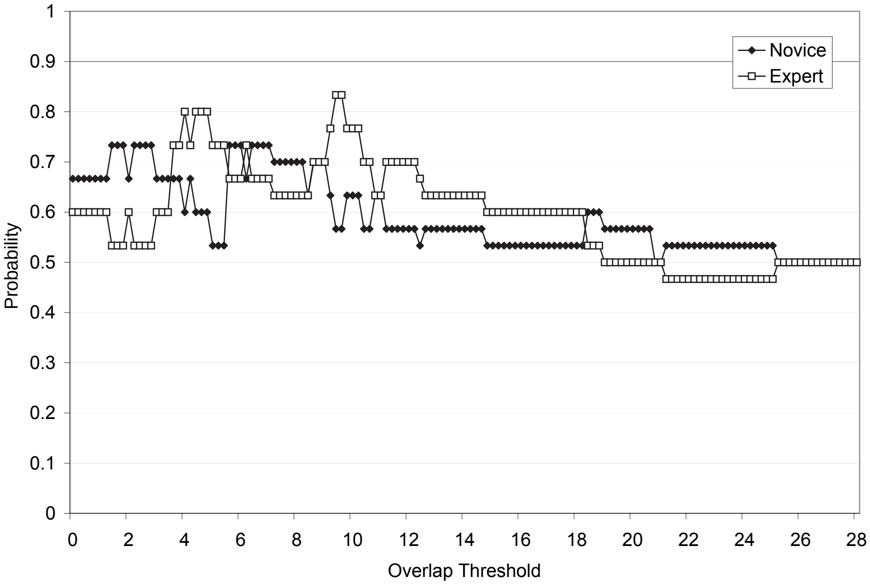


Figure 6: Probability of predicting the correct ordering (novice-like or expert-like) averaged over the dataset, against threshold, for Karla the Hawk data. Features of intermediate size appear to figure in expert behavior, but with a less dramatic effect than seen in the Conflict dataset.

3.3.2 Results

The results, shown in Figure 6, show a modest effect that is consistent with our expectations. The results are noisier and not as clear as those in Experiments 1 and 2, and we hypothesize that this is so for two reasons: first, the lack of a class of less-analogical sources in the Karla data, and second, the less dramatic nature of the analogical relationships in the Karla data (less dramatic relative to those in our generated dataset). These points are discussed in more detail in Section 4.1.

[7] Story sets 5, 7, 8, 12, and 17.

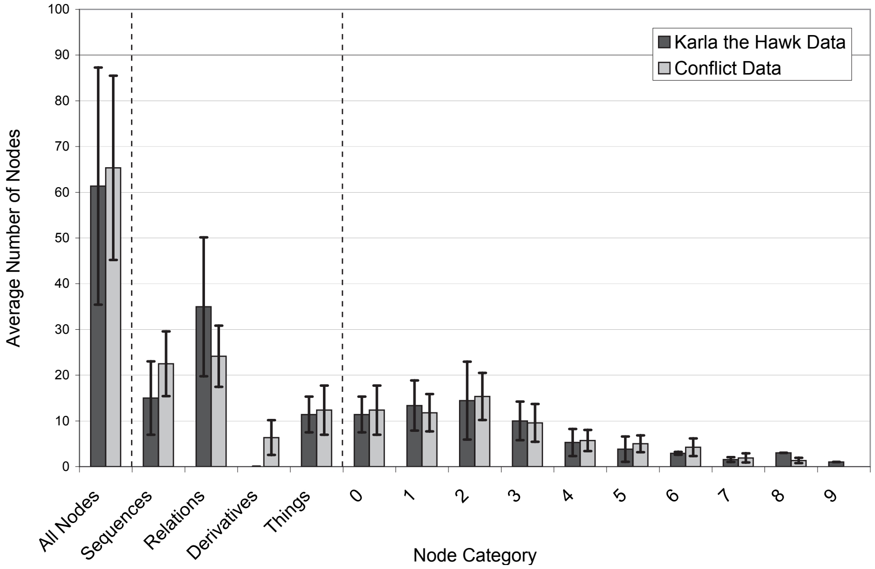


Figure 7: Histogram of the average number of nodes in different categories. Error bars indicate one standard deviation. The regions between the dashed lines indicate different ways of characterizing all the nodes in the descriptions: the first bar deals with all nodes; the second through fifth bars deal with nodes divided by type; the sixth through fifteenth bars (those numbered 0 to 9) indicate nodes broken down by order. The Karla the Hawk data is only for the five scenarios used in our experiments. The Karla the Hawk data does not have a node category analogous to what we call derivatives. Our Conflict dataset and this subset of the Karla the Hawk dataset exhibit similar node counts in various categories and at various sizes.

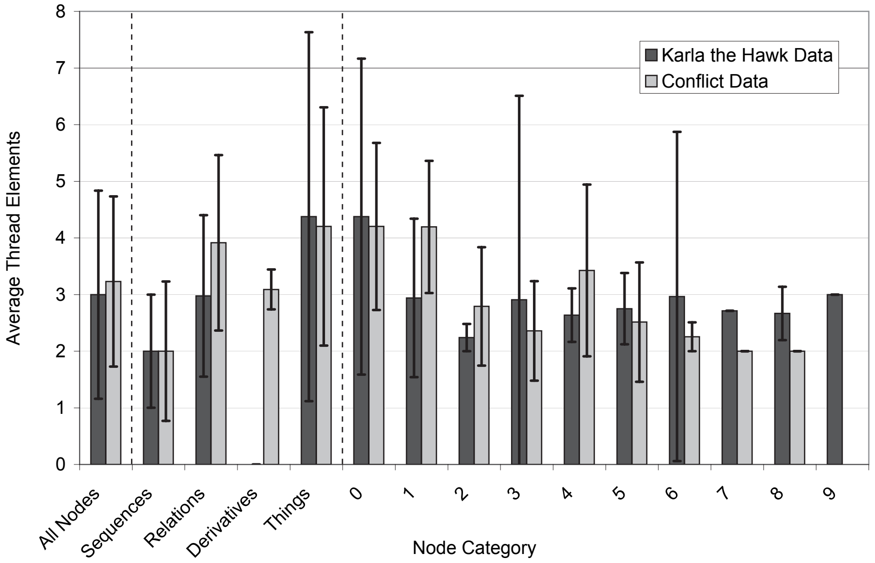


Figure 8: Histogram of the average number of semantic elements (thread elements) per node for different categories. Error bars and labels are the same as in Figure 7. Evidently, our Conflict data and the Karla the Hawk data exhibit similar characteristics with respect to the number of classes associated with nodes in various categories and at various sizes.

4 Discussion

4.1 Experimental Results

Here we review the results of the three experiments and the interrelationships of those results. The retrieval model works by breaking the target into features (subgraphs, or pieces of the description), and then looking for these features in the sources to determine which sources are more likely to be retrieved. If one assumes that the most informative features are, on average, those of intermediate size, the most obvious way to bias toward expert-like performance is to filter on feature size directly, allowing only intermediate features to influence retrieval. Contrary to this expectation, the model does not operate this way. Instead it filters of feature-feature match score. Why was the model designed this way?

The short answer is that the datasets are too small to show an effect if the model filters directly on feature size. If the model filtered on feature size alone, it would allow both informative and uninformative intermediate features to affect retrieval, and, with such a small dataset, this lack of specificity in the informativeness of the features leads to losing the expert-like retrieval effect in the noise. One would expect that with an extremely large dataset, say, a thousand, or ten-thousand precedents, analogous matches would rise above the non- or less-analogous matches to the extent which intermediate features are on the average more informative; but because the difference in mutual information is relatively small, the noise dominates with a small dataset.

As noted in the next section, to be truly effective using intermediate features for retrieval, you have to know what features are most informative before you begin. In our experiments, we must estimate the mutual information of the features because we do not have a large dataset that allows us to measure it directly. The feature-feature match score is an estimate of how informative the feature is, convoluted with how large it is. Thus, instead of filtering on the size of the features, as is the obvious way of directly demonstrating the effect, we filter on how well the feature from the target matches the source. The feature-feature match score is loosely analogous to the informativeness of the feature for that particular source. Match scores of intermediate magnitude correspond to good matches between pieces of intermediate sized structure, be they features of an intermediate size or portions of larger features. Thus, when we filter on match score, to only allow those of intermediate magnitude to effect the retrieval, this will have the effect of biasing the retrieval toward highly-informative intermediate features, and will appropriately test the hypothesis.

Having understood the design of the model, we are now in a position to examine the results of Experiments 1 and 2. In Figure 9 the results of both experiments are placed side-by-side, with Experiment 2 on the left and Experiment 1 on the right. The thresholds in the two experiments control the range of match scores that are included in the retrieval score of each source. At the center of the figure, where the two graphs meet, match scores of all magnitudes influence the retrieval score. This gives novice-like retrieval. As we move to the left, the upper threshold of Experiment 2 removes higher-scoring matches first. When we reach the point where we are only allowing the smallest match scores to affect retrieval (as we approach the far left of the figure), every match score is a large portion of the retrieval, and so as they are slowly removed, the scores vary wildly and unpredictably. Starting in the middle again, and moving toward the right side of the figure, the lower threshold starts from zero and steadily increases, throwing away smaller scoring matches first. As we move to the far right of the figure, only high-scoring matches affect retrieval, and since these are extremely rare, most of the retrieval scores are zero.

The overall effect of the two thresholds can be thought of as a 'window' that moves from left to right over the range of possible match scores. Wherever the window falls, these match scores are added together to form the retrieval score. This is indicated by the boxes at the bottom of the figure. The order of the retrieval scores for each target's generated source matches is then used to calculate the fraction of times that the retrieval algorithm produces the expert-like or novice-like retrieval order at that threshold. As can be seen, the only place where expert-like retrieval is achieved is when the lower threshold is slightly less than an intermediate size, meaning that both intermediate and large feature match scores are contributing to the retrieval scores. Because large feature match scores do not achieve expert-like retrieval (the far right side of the graph), and because, by virtue of the retrieval score being a linear sum, the inclusion of large features neither suppresses nor amplifies the effect of other features, we must conclude that intermediate features alone are responsible for the expert-like retrieval effect.

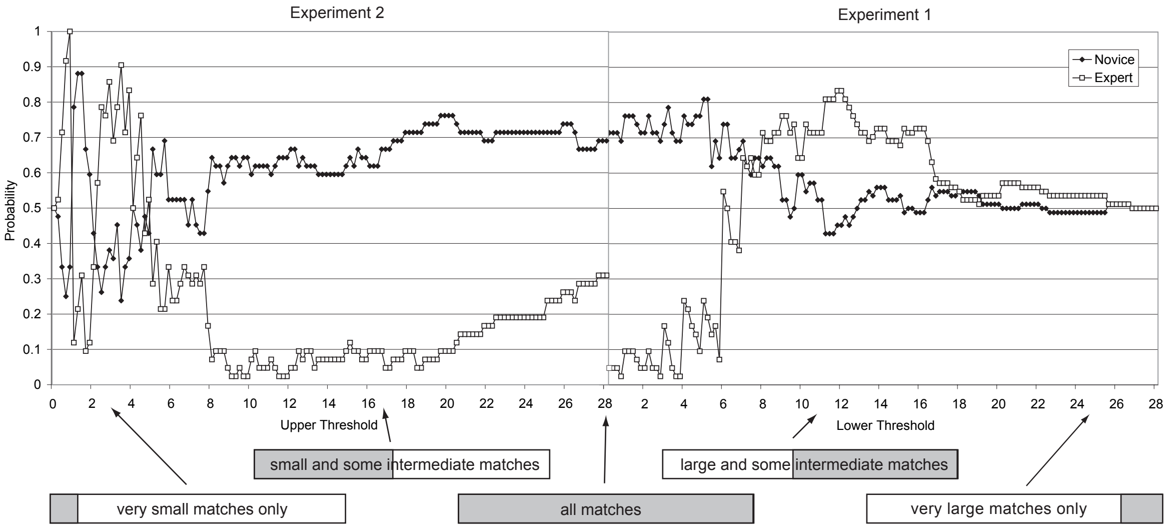


Figure 9: The range of features match scores that are included in the retrieval procedure for various values of the lower and upper thresholds. The retrieval procedure always uses a continuous interval of match scores; the interval is bounded below by the lower threshold, and above by the upper threshold. This range is represented by the boxes at the bottom of the diagram. At the far left only small feature matches are used. In the middle all matches are used. At the far right only large matches are used.

Because we see only expert-like retrieval when we cut off the bottom-third of the match score range, we repeated only Experiment 1 on the Karla the Hawk dataset. The Karla the Hawk data set is also small, only nine stories; because we took care to choose a subset of the stories that produced similar statistics in the number of nodes and number of semantic tags, our sample was reduced to five stories. Even so, there is a noticeable expert-like retrieval effect at an intermediate size.

A few words on the modest size of the expert-like retrieval effect with the Karla the Hawk stories. The dataset was designed for dual use in both computational and psychological experiments, and so the associated analogies are much more subtle than those generated by our source synthesis procedure. Because, in the Karla the Hawk dataset, there is only a small distinction between a mere-appearance source and an analogical source, noise due to small sample size causes the probability curves to jump back and forth erratically as features are removed from consideration and the mere-appearance and analogical retrieval scores rise and fall. We believe that this could have been solved by the introduction of a less-analogical source into the dataset. This is illustrated in Figure 10. The expert-like and novice-like order calculations depend on the position of the mere-appearance source relative to all the other sources. If there had been a a less-analogical target (or better yet, a series of less-analogical targets running the range from non-analogical to analogical), any particular order of the mere-appearance source relative to any other source would be a much smaller fraction of the average, and so slight variations at the transition point (i.e., the threshold), will have less effect.

In addition to this sampling effect, the Karla the Hawk results are more sensitive to variation in scenario size because of the subtlety of the relationships. This is because 'intermediate' is relative to the size of the scenario. The larger the scenario, the larger the size of an intermediate feature. Although the scenario sizes vary widely in the Conflict dataset, we did not have to normalize to see an effect. This is because the analogical relationships were quite dramatic, and there was some overlap in the intermediate features range. In the Karla the Hawk data, we believe the distribution of scenario sizes contributed a suppression of the expert-like effect. In future experiments, we may wish to normalize for scenario size and thereby show a more dramatic effect in the Karla the Hawk data.

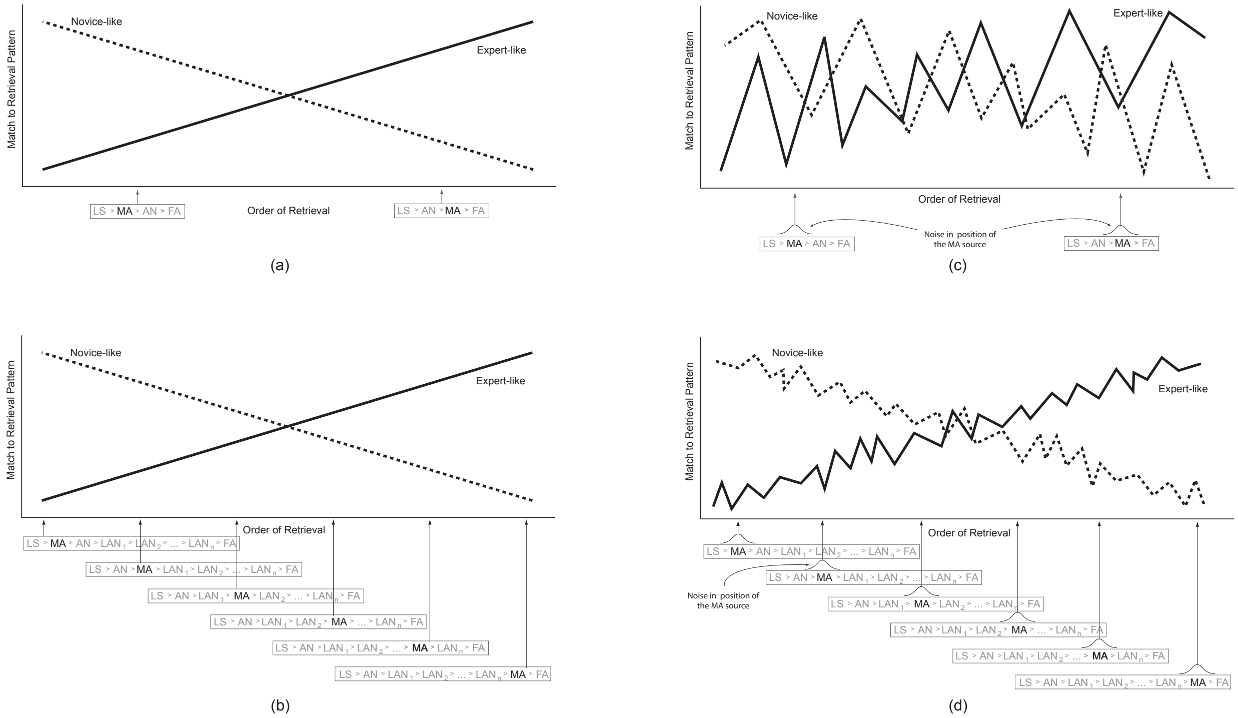


Figure 10: Graphs of the match of the expected performance of the model to novice-like and expert-like behavior versus the order of LS, MA, AN, and LAN sources. (a) and (b) are ideal cases, when all scenarios contain infinitely many features of all sizes, resulting in perfectly smooth lines. In these cases, it doesn't matter if one has less-analogical matches (LANs) or not. (c) and (d) show the non-ideal cases, when scenarios are made up of a limited number of features. In (c), where there are no less-analogical matches, any noise in the position of the MA match can change the measured match to the novice-like or expert-like order by up to 1/3. In (d), where there are n less-analogical matches of varying strengths, noise can only change the measured match to the novice-like or expert-like order by $1/(3+n)$. Thus the variance of the noise in case (d) is smaller.

4.2 Expertise and Intermediate Features

Our hypothesis concerns the utility of intermediate features for producing expert-like analogical retrieval. We do not claim that our algorithm represents what actually occurs *in the brain*, but rather that it indicates information that must computed at some point in the process. We interpret our results in a framework akin to those presented in (Keane, Ledgeway, & Duff, 1994) and (Palmer, 1989). In these views, the ACME and SME models of analogy provide *computational*-level constraints on the process of analogy, and Keane's IAM model provides *algorithmic*-level constraints. Similarly, we see this work as indicating a computational-level constraint on the process of retrieval: the human retrieval system attempts to extract intermediate features from targets and sources for use in retrieval.

This being said, we have demonstrated that, in principle, the use of intermediate features in retrieval can lead to expert-like performance. Is this to say that intermediate features are alone responsible for expertise? This would be a strong claim; we do not make this claim, for at least two reasons.

First, consider the following piece of (specious) advice: Because intermediate features allow you to perform like an expert, just look for intermediate features when you are retrieving, and you will become an expert. Clearly this advice borders on the absurd. Turning a novice into an expert requires more than just this. What more does it require? It requires that you know which intermediate features to look for. We can see this by thinking of the many different ways of describing any particular scene. Take the example from above, the scene described by "The boy ran to the rock and hid because the girl threatened him." If we were actually watching this scene take place, how might we possibly describe it? There are innumerable ways, only some of which can be used for identifying useful analogies. For example, we might take note of the girl's threatening words, and how the boy was crouching behind the rock and peeking out from behind it. Alternatively, we might go into detail about what they were wearing, or how exactly their limbs moved relative to their bodies in causal or temporal sequence, or describe the series of rationalizations and emotions that the boy and girl experienced as the scene played out. All of these might be valid descriptions, and in them we could find features of an intermediate size. But they would not be useful for finding analogs like "The army retreated to the castle and dug in because the enemy army approached." There are an extremely large number of possible descriptions of any scene, and an extremely large number of intermediate features for each of those descriptions, and only some of these intermediate features are highly informative. Thus the qualification, found throughout this paper, that intermediate features are only *on the average* more informative than features of other sizes. Not all intermediate features are highly informative (and, indeed, not all small or large features are uninformative). What it takes to make a novice into an expert is not only the knowledge that one should be looking for features of an intermediate size, but the knowledge of which ones are the most informative.

Second, as some readers may have already realized, there are some cases where describing the scene in terms of the clothes worn, the movement of the limbs, or the emotions experienced would allow the describer to discover useful analogs. That is the case where you are interested in precedents that are analogous *in that way.* In this case, all the scenes (or some large, relevant fraction) in your memory would need to be described in that way, and people with these sorts of memory we would probably consider 'experts' in fashion design, or biomechanics, or psychology, respectively. This leads to the second reason that intermediate features cannot be the sole mechanism of expertise: expertise requires *representational uniformity.*[8] If you are not casting descriptions in the same sorts of representations, with the same descriptive conventions, highly informative intermediate features from one domain and type of description will not help you find useful analogs in your memory, since the precedent that would help is probably not coded in the right form.

These two reasons that intermediate features cannot be the sole mechanism of expertise raise two important questions: how do experts learn which features are the most informative, and how do they learn what representations most efficiently reveal these features? This is an important question, and an answer is required for the proper integration of intermediate features into practical models of retrieval (as discussed in the next section). However, it is not a topic we tackle in this current work, and thus will say no more.

4.3 Leveraging Intermediate Features

If we take for granted the result that intermediate features are the most useful features for analogical retrieval, how could we leverage this knowledge in a practical retrieval scheme? As already noted, the algorithm used in this work was designed to circumvent the small size of the available datasets. In this section we propose a straightforward way of using intermediate features in retrieval. We also indicate how the intermediate features insight can be integrated into two other prominent models of analogical retrieval, the MAC/FAC model and the ARCS model, to allow those models to produce novice-like or expert-like retrieval.

It is important to note first that all of these approaches to using intermediate features presuppose that the most informative set of features (most of them intermediate) is already known. This is major presupposition, and, as mentioned in the previous section, it is not a topic that we treat in this work. Presumably this process of discovery is time- and resource-intensive, and the set of features discovered domain-specific. Before intermediate features can be practically integrated into a model of retrieval an adequate solution must be presented to the problem of discovering the most informative set of features for retrieval.

A simple procedure for leveraging a known set of intermediate features might be as follows. First, as precedents are learned, they are analyzed in light of those features that are most informative for the domain.

[8] This has already been proposed by a number of other research. See, for example, (Chi et al., 1981), or Forbus and colleagues in (Forbus et al., 1994, Section 7.1.7) as a way to extend MAC/FAC model of analogical retrieval to cover expert-like results, viz. richer and better structured representations and greater representational uniformity.

The precedents are then indexed by those features, allowing them to be retrieved with little or no effort. Specifically:

1. Analyze potential precedents at storage time, using a $O(n^2 \log n)$ greedy matcher (in the number of nodes) to look for features of an intermediate size that are known to be popular in the domain.

2. Characterize each stored precedent by the names of the features contained.

3. Retrieve precedents using named features and an $O(n)$ vector dot-product method (in the length of the vector).

For those readers familiar with other models of analogical retrieval, this above proposed method will likely strike them as similar to the MAC/FAC model by Forbus, Gentner, and Law (1994). This is no accident. While our model stands apart from MAC/FAC by virtue of its variable feature-size mechanism, and MAC/FAC has not yet been used to account for the distinction between novice-like and expert-like retrieval, we believe that our model has strong intersections with MAC/FAC. MAC/FAC is a two-stage process: the first stage, the "Many Are Called" stage (MAC), performs a computationally inexpensive non-structural match between a target and source in memory; the second, "Few Are Chosen" (FAC), stage carries out a more expensive structural match between the target and the best source candidates from the first stage. The non-structural match score (the MAC stage) is the dot-product between two *content vectors*. These vectors are a non-structural summaries of the numbers of objects and relations found in that description. For example, if a description is "The brown dog bit the old dog and the nice man," the content vector would be `[brown=1, dog=2, bite=1, man=1, old=1, nice=1]`. The more computationally expensive FAC stage operates on the best results returned by the MAC stage, using a structural mapper to score them relative to the target according to the soundness of their structural mappings. The process uses two thresholds, one to choose the top portion of the MAC stage output for input to the FAC stage, and one to choose the top portion of the FAC stage output for retrieval results. MAC/FAC was used to model several psychological phenomena: the primacy of literally-similar remindings, the high frequency of mere-appearance similar remindings, and the rare occurrence of structurally-based (analogical) remindings. The model produces what we have been calling the novice-like retrieval pattern: literally-similar > mere-appearance > analogical.

Integrating a variable feature-size mechanism in MAC/FAC could be accomplished by an extension of the MAC stage. As it stands, the MAC content vector includes only small features: individual objects and relations. Thus, consistent with our results, MAC/FAC achieves novice-like retrieval. If the MAC stage were allowed to include pieces of description of all sizes on the content vector, and the dot-product operation was extended to allow structural comparison between the pieces, this would allow one to vary the informativeness of the description pieces that influence the retrieval. If the most informative features were known for the domain, and it was made sure that these were added to the content vector, then, we hypothesize, the retrieval of the model would be tuned to expert-like retrieval. If this were so, the model could then be tuned to any level in between novice and expert by varying the numbers or weights of the various sizes of features. We suspect that this inclusion of structure in the MAC stage would eliminate or reduce the need for the FAC stage.

Intermediate features can also inform the other major model of analogical retrieval, the ARCS model by Thagard, Holyoak and colleagues (Thagard et al., 1990). ARCS, which stands for Analog Retrieval by Constraint Satisfaction, effects retrieval by creating a constraint-satisfaction network over correspondences between representational pieces of the source and the target, then using a standard parallel connectionist relaxation algorithm to allow the network to settle into a state which indicates the relative strength of retrieval of various targets in memory. The nodes in the network are constructed by applying semantic constraints between the source and targets, and links (either excitatory or inhibitory) are constructed using systematic and pragmatic constraints. Like the MAC/FAC model, ARCS models only a single type of retrieval that is, as far as can be told, somewhere between novice-like and expert-like. Intermediate features point the way to extending ARCS to model both novice-like and expert-like retrieval. We hypothesize that if a filter were applied to the construction of nodes in the constraint network, biasing toward groupings of nodes that were more informative, the network could be tuned to novice-like or expert-like retrieval.

4.4 The Goldilocks Hypothesis

We see this work as supporting a broad hypothesis that we call the *Goldilocks hypothesis*. This hypothesis states that, on the average, those features that are maximally useful for recognition are neither too small nor too big, neither too simple nor too complex, but rather in the middle: intermediate in size and complexity. Taken as stated, this claim is quite broad, and is not limited to a specific domain or representation, but instead applies to representation *in general*. Our work could be considered a single instance of support for this hypothesis; taken with other instances, it is suggestive, although by no means conclusive, of the hypothesis's computational power. We have already reviewed a second instance that we see as supporting the hypothesis, that of Ullman's computational work in vision. In this section we will highlight two additional areas of study that we think speak to the validity of the Goldilocks hypothesis. We hope that presenting these beside one another suggests, more than mere speculation might, that the Goldilocks hypothesis has some reality.

The first results beyond our own work and that of Ullman's that we think support the Goldilocks hypothesis are from the study of narrative structure. Since the very early days of Artificial Intelligence and Cognitive Science, there has been an interest in characterizing the representations that are used to understand and store narratives, whether gathered from discourse or from one's own experience (e.g., autobiographical). There has been a focus on what has variously been called *scripts*, *plans*, *schemata*, or *themes* as representational structures. For example, one of the very earliest attempts to set out a theory of narrative representation, that of *plot units*, was by Lehnert (1981). Conceiving of stories as most profitably characterized at a level of abstraction that would include such events as *retaliation* or *regrettable mistake*, Lehnert and colleagues showed how these representations could distinguish between stories, and map out their flows and their chokepoints (Alker, Lehnert, & Schneider, 1985).

Experimental investigation into the psychological reality of plot units indicated that people were indeed highly sensitive to plot units and protagonist goals (Reiser, Black, & Lehnert, 1985). It has been found that the accurate recall of a story is a function of its inherent plot structure (Thorndyke, 1977), and that higher-order elements in a story (as indicated by a hierarchical description of the story) are remembered better over long periods of time (Stanhope, Cohen, & Conway, 1993). Perhaps the most convincing demonstration of the psychological reality of characterizations of stories at an intermediate level have come from psychological experiments that demonstrate that thematically similar stories can show priming effects without inducing actual recall of a similar story (Seifert, Abelson, McKoon, & Ratcliff, 1986; Schunn & Dunbar, 1996). This strongly suggests that thematic representations are actively extracted and explicitly stored in memory. Additional evidence for intermediate features in coherent narratives comes from the rich literature on the study of autobiographical memories. It has been shown, for example, that old autobiographical memories are organized by theme, or distinctive element (Sato, 2002), and that activities are preferred for indexing of autobiographical memories (Reiser, Black, & Abelson, 1985). There has been suggestive work that events serve as a strong organizer for autobiographical memories (N. R. Brown & Schopflocher, 1998).

These structures can be conceived as intermediate representations that maximize one's ability to recognize and categorize different sorts of narratives. Note that here there is a direct interplay between feature size and abstraction. Abstraction can be seen as a method for generating intermediate (and larger) features and vice-versa: intermediate features can be seen as abstractions. The actual formal size of the feature is not important; what is important is the mutual information of the feature for the category that the feature is used to recognize. Abstractions, and similarly features larger than those of atomic size, are constructed by integrating together pieces of information about what is being described. The Goldilocks hypothesis states that, at some point, on the average, you reach a point of diminishing returns: the more pieces of information you add to your description, or the more cases you cover by abstracting away detail, the less mutual information you get.

We move to our fourth and final instance in support of the Goldilocks hypothesis, in the domain of object categorization. In the mid-1970's Rosch proposed that object prototype effects were the result of a principle of cognitive economy which put the maximum amount of information in a privileged middle level of categorization hierarchies (Rosch, 1978). She proposed that there is a level of abstraction, called the basic level, which lies in between superordinate categories (which include many objects), and specific categories (which include few objects). Experiments have repeatedly shown the primacy of basic-level categories in identification tasks (Murphy & Smith, 1982; Murphy, 1991), and other researchers have extended her work to other domains, such as events (Rifkin, 1985; Morris & Murphy, 1990). This work in psychology was complemented by related work in anthropology which demonstrated that many cultures classify living things in a hierarchy centered on what we call the genus level, which provides more knowledge than other levels, both lower and higher (Berlin, Breedlove, & Raven, 1973).

Why is the basic level is privileged? The reasoning is that concepts at the basic level provide “maximum information with the least cognitive effort” (Rosch, 1978). This hypothesis has been lent experimental support by the work of Tversky and colleagues (1984), who showed that goal- and part-related information, the kind most informative for the use of the object, is usually encoded at the basic level, while object attributes are usually found at a lower, more specific level.

As the reader has no doubt noticed, the basic level has many similarities with the intermediate level. The argument here is the same as for narrative representation: the basic level is preferred because it is the level where one finds the right amount of collation and smoothing of descriptions, in other words, the right amount of abstraction. The Goldilocks hypothesis can cover the basic level in the same way it covers features at the intermediate level: they are preferred for identification and categorization because they maximize the mutual information for the class.

5 Contributions

Here we articulate what we believe to be the contributions of our work.

First, we identified the distinction between novice-like retrieval (retrieval by mere-appearance) and expert-like retrieval (retrieval by structural congruence) in analogical problem solving tasks, as a yet unmodelled aspect of human performance.

Second, we hypothesized that we can parsimoniously account for the distinction between novices and experts by way of a single cognitive mechanism, that of searching sources in memory for relevant features. With this in mind we demonstrated a model based on that mechanism that varied the size of the descriptive features permitted to influence the judgment of similarity. Using small features inclined the model to novice-like performance; using features of intermediate size inclined the model to expert-like performance. We concluded that the difference between novice-like and expert-like retrieval corresponds not only to the development of procedures that identify maximally informative features, but also to the development of more uniform representational schemes.

Third, we described how the insight of intermediate features could be realized in a more practical retrieval model, or integrated into existing models of analogical retrieval. We concluded that integrating their use into current and future retrieval models would require a process for learning the most informative features.

Fourth, and finally, we introduced the Goldilocks hypothesis, which says that for identification in general, features of an intermediate size are best because they, on average, maximize the mutual information. We call it the Goldilocks hypothesis because it reminds us of the fairy tale, *Goldilocks and the Three Bears*, in which Goldilocks found herself most satisfied with porridge that was neither too hot nor too cold. We noted that the Goldilocks hypothesis can be supported by both our work in precedent retrieval and the work of Ullman and his colleagues in the world of object recognition. We further observed that the Goldilocks hypothesis might be supported by work in other cognitive domains, specifically narrative memory and object categorization. Such examples suggest to us that the Goldilocks hypothesis contains a powerful computational idea that is exploited in a broad range of human cognitive faculties.

6 Acknowledgments

Thanks to all the members of our research group for their assistance and constructive critiques, especially Jake Beal, Keith Bonawitz, Harold Cooper, Ian Eslick, Roger Hurwitz, Paul Keel, Ben Lamothe, Vikash Mansinghka, John Mallery, Diana Moore, Sajit Rao, and Mark Seifter. Thanks to Ken Forbus for providing the Karla the Hawk dataset and his helpful advice, as well as to his Qualitative Reasoning Group at Northwestern for their comments and suggestions. Thanks to Liz Bonawitz and Amy Perfors of the Tenenbaum laboratory in the MIT Department of Brain and Cognitive Sciences for their help and advice. Thanks to Arthi Murugesan for preparing the code which generated Figures 7 and 8. Portions of this work appeared as a poster entitled "A Model of Analogical Retrieval using Intermediate Features" at the Twenty-Sixth Annual Meeting of the Cognitive Science Society, 2004, in Chicago, Illinois, and as a paper entitled "Intermediate Features and Informational-level Constraint on Analogical Retrieval"at the Twenty-Seventh Annual Meeting of the Cognitive Science Society, 2005, in Stresa, Italy. This work was supported in part by the National Science Foundation under grants IIS-0218861 and IIS-0413206.

References

Alker, H., Lehnert, W., & Schneider, D. (1985). Two reinterpretations of toynbee's jesus: Explorations in computational hermeneutics. In G. Tonfoni (Ed.), *Artificial intelligence and text understanding* (p. 29-94). Quaderni di Ricerca Linguistica.

Berlin, B., Breedlove, D. E., & Raven, P. H. (1973). General principles of classification and nomenclature in folk biology. *American Anthropologist*, *75*, 214-242.

Blessing, S. B., & Ross, B. H. (1996). Content effects in problem categorization and problem solving. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*(3), 792-810.

Brown, A. L. (1989). Analogical learning and transfer: What develops? In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (p. 369-412). Cambridge: Cambridge University Press.

Brown, N. R., & Schopflocher, D. (1998). Event cueing, event clusters, and the temporal distribution of autobiographical memories. *Applied Cognitive Psychology*, *12*, 305-319.

Chi, M. T. H., Feltovich, P. J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, *5*, 121-152.

Ciment, J. (1999). *Encyclopedia of conflicts since world war ii.* M.E. Sharpe.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1986). The structure-mapping engine. In *Fifth meeting of the american association for artificial intelligence* (p. 272-277).

Faries, J. M., & Reiser, B. J. (1988). Access and use of previous solutions in a problem-solving situation. In *Annual conference of the cognitive science society* (Vol. 10, p. 433-439). Montreal: Lawrence Erlbaum and Associates.

Forbus, K. D., Gentner, D., & Law, K. (1994). Mac/fac: A model of similarity-based retrieval. *Cognitive Science*, *19*, 141-205.

French, R. M. (2002). The computational modeling of analogy-making. *Trends in Cognitive Sciences*, *6*, 200-205.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*, 155-170.

Gentner, D., & Landers, R. (1985). Analogical reminding: A good match is hard to find. In *International conference on systems, man and cybernetics* (p. 607-613). Tucson, AZ: IEEE.

Gentner, D., Rattermann, M. J., & Forbus, K. D. (1993). The roles of similarity in transfer: Separating retrievability from inferential soundness. *Cognitive Psychology*, *25*(4), 524-575.

Gick, M. L., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology*, *12*, 306-355.

Gick, M. L., & Holyoak, K. J. (1983). Scheme induction and analogical transfer. *Cognitive Psychology*, *15*, 1-38.

Hall, R. P. (1989). Computational approaches to analogical reasoning: a comparative analysis. *Artificial Intelligence*, *39*, 39-120.

Holyoak, K. J., & Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory and Cognition*, *15*, 332-340.

Jackendoff, R. (1983). *Semantics and cognition.* MIT Press.

Keane, M. T., Ledgeway, T., & Duff, S. (1994). Constraints on analogical mapping: A comparison of three models. *Cognitive Science*, *18*, 387-438.

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by.* University of Chicago Press.

Lehnert, W. (1981). Plot units and narrative summarization. *Cognitive Science*, *4*, 293-331.

Lowe, D. G. (2000). Towards a computational model for object recognition in it cortex. In S.-W. Lee, H. Bulthoff, & T. Poggio (Eds.), *Workshop on biologically motivated computer vision* (Vol. 1811). Springer-Verlag.

Morris, M., & Murphy, G. L. (1990). Convering operations on a basic level in event taxonomies. *Memory and Cognition*, *18*, 107-118.

Murphy, G. L. (1991). Parts in object concepts: Experiments with artificial categories. *Memory and Cognition*, *19*, 423-428.

Murphy, G. L., & Smith, E. E. (1982). Basic level superiority in picture categorization. *Journal of Verbal Learning and Verbal Behavior*, *21*, 1-20.

Novick, L. R. (1988). Analogical transfer, problem similarity, and expertise. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *14*, 510-520.

Novick, L. R., & Holyoak, K. J. (1991). Mathematical problem solving by analogy. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *17*(3), 398-415.

Nyberg, E. H., & Mitamura, T. (1996). Controlled language and knowedge-based machine translation: Principles and practice. In *First international workshop on controlled language applications.* Katholieke Universiteit, Leuven.

Obeid, M., Jedynak, B., & Baoudi, M. (2001). Image indexing and retrieval using intermediate features. In *Association for computing machinery international conference on multimedia* (Vol. 9, p. 531-533). Ottawa: Association for Computing Machinery.

Palmer, S. E. (1989). Levels of description in informatino-processing theories of analogy. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (p. 332-345). Cambridge: Cambridge University Press.

Rattermann, M. J., & Gentner, D. (1987). Analogy and similarity: Determinants of accessibility and inferential soundness. In *Annual conference of the cognitive science society* (Vol. 9, p. 23-35). Lawrence Erlbaum Associates.

Reed, S. K., Dempster, A., & Ettinger, M. (1985). Usefulness of analogous solutions for solving algebra word problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *11*, 106-125.

Reed, S. K., Ernst, G. W., & Banerji, R. (1974). The role of analogy in transfer between similar problem states. *Cognitive Psychology*, *6*, 436-450.

Reiser, B. J., Black, J. B., & Abelson, R. P. (1985). Knowledge structures in the organization and retrieval of autobiographical memories. *Cognitive Psychology*, *17*, 89-137.

Reiser, B. J., Black, J. B., & Lehnert, W. G. (1985). Thematic knowledge structures in the understanding and generation of narratives. *Discourse Processes*, *8*, 357-389.

Rifkin, A. (1985). Evidence for a basic level in event taxonomies. *Memory and Cognition*, *13*, 538-556.

Rosch, E. (1978). Principles of categorization. In E. Rosch & B. Lloyd (Eds.), *Cognition and categorization* (p. 27-48). Hillsdale, NJ: Lawrence Erlbaum Associates.

Ross, B. H. (1984). Remindings and their effects in learning a cognitive skill. *Cognitive Psychology*, *16*, 371-416.

Ross, B. H. (1987). This is like that: The use of earlier problems and the separation of similarity effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *13*, 629-639.

Sato, K. (2002). Changes in the temporal organization of autobiographical memories. *Psychological Reports*, *91*, 1074-1078.

Schoenfeld, A. H., & Herrmann, D. J. (1982). Problem perception and knowledge structure in expert and novice mathematical problem solvers. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *8*, 484-494.

Schunn, C. D., & Dunbar, K. (1996). Priming, analogy, and awareness in complex reasoning. *Memory and Cognition*, *24*, 271-284.

Seifert, C. M., Abelson, R. P., McKoon, G., & Ratcliff, R. (1986). Memory connections between thematically similar episodes. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *12*, 220-231.

Serre, T., Riesenhuber, M., Louie, J., & Poggio, T. (2002). On the role of object-specific features for real world object recognition in biological vision. In H. Bulthoff, S.-W. Lee, T. Poggio, & C. Wallraven (Eds.), *Second workshop on biologically motivated computer vision; in lecture nodes in computer science* (Vol. 2525, p. 387-397). Springer-Verlag.

Shneiderman, B. (1977). Measuring computer program quality and comprehension. *International Journal of Man-Machine Studies*, *9*, 465-478.

Stanhope, N., Cohen, G., & Conway, M. (1993). Very long-term retention of a novel. *Applied Cognitive Psychology*, *7*, 239-256.

Thagard, P., Holyoak, K. J., Nelson, G., & Gochfeld, D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence*, *46*, 259-310.

Thorndyke, P. W. (1977). Cognitive structures in comprehension and memory of narrative discourse. *Cognitive Psychology*, *9*, 77-110.

Tversky, B., & Hemenway, K. (1984). Objects, parts, and categories. *Journal of Experimental Psychology: General*, *113*, 169-193.

Ullman, S., Vidal-Naquet, M., & Sali, E. (2002). Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, *5*, 682-687.

Vaina, L., & Greenblatt, R. (1979). *The use of thread memory in amnesia aphasia and concept learning* (Working Paper No. No. 195). MIT Artificial Intelligence Laboratory.

Winston, P. H. (1980). Learning and reasoning by analogy. *Communications of the ACM*, *23*, 689-703.

Winston, P. H. (1982). Learning new principles from precedents and exercises. *Artificial Intelligence*, *19*, 321-350.

A Appendix: Implementation Details

In this section we explain our model in enough detail to allow its re-implementation. We begin by describing our representational conventions, and discuss how we break descriptions into features. We review the actual matching algorithm and analyze its computational complexity. Finally, we give examples of calculations of node-to-node semantic similarity scores and feature-to-feature match scores.

A.1 Representations

The actual representational data structures on which the model algorithm is run are quite similar to others used in research on analogy (Thagard et al., 1990). The representation encompasses both episodic memory (implemented as a graph), and semantic memory (implemented as frames attached to nodes in the graph).

As is often found in graph representations such as these, nodes in the graph represent objects or concepts, and the edges that connect the nodes represent a simple relationship between concepts or objects. In our representation, we have three edge types: *subject*, *object* and *element*. These names are motivated by the examples in Figure 11, where the 11(a) represents the sentence "Amy loves Bob"; because the node representing *Amy* is the subject of the verb *Loves*, the link connecting them is of the *subject*. Similarly for *Bob* as the object of the sentence. The graph in 11(b) shows an example of the use of the *element* edge type, where the graph is representing the concept "A and B are members of C." This construction is used to collect similar nodes into collections where *subject* and *object* labels don't apply. As a convention, we say that particular edges *belong* or *originate* from only one node. This is identical to saying that the edge is *unidirectional*. So in 11(a), the subject and object edges belong to the *Loves* node, and in 11(b), the element edges belong to the *C* node.

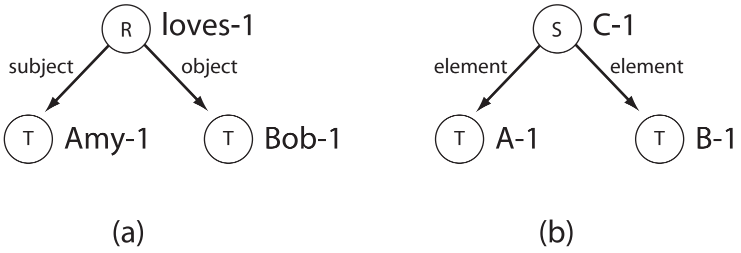


Figure 11: Representations of (a) the sentence "Amy loves Bob," and (b) the concept "A and B are members of C." Note that each node has a type that is indicated by a letter inside of the node.

We have four different node types that are loosely based on the linguistic function-argument structure presented in Jackendoff (Jackendoff, 1983), and are distinguished by how many and what type of edges originate on that node. The simplest sort of node has no originating edges–we call this a **Thing** node, which usually represents an object or simple concept. Examples include 'dog', 'boy', 'man', 'rock', 'tree', etc.

The next simplest node is one that originates only one edge (by convention, a *subject* edge; no node originates only an *object* edge). These nodes usually indicate something like a preposition, such as "on the table", or "top of the tree." Because the meaning of these nodes derives in some sense from their subjects, we call these nodes **Derivatives**, and these examples are illustrated in Figure 12.

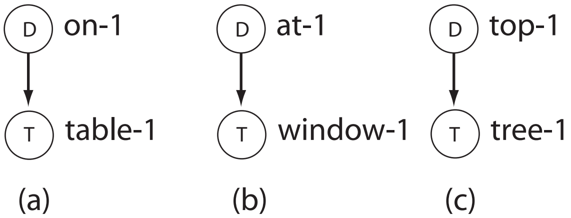


Figure 12: Representations of the phrases (a) "on the table," (b) "at the window," and (c) "top of the tree."

Nodes that originate both a subject and object link are called **Relations** and represent relations such as *Causes* or *Loves*, as was shown in 11(a). Finally, there are nodes, called **Sequences**, from which originate one or more element links (with no subject or object links). These represent collections of events or locations, and can be treated as ordered or unordered. An example is shown in 11(b). A summary of the possible node types and their characteristics is shown in Table 2.

Type	subject?	object?	elements?	order	Examples
Thing	No	No	No	0	boy, tree, dog, rock
Derivative	Yes	No	No	max(children)+1	in, out, to, from
Relation	Yes	Yes	No	max(children)+1	is, loves, before, causes
Sequence	No	No	Yes	max(children)+1	paths, conjunction, scenarios

Table 2: Ontology of node types

We follow conventions listed in (Gentner, 1983) in assigning an *order* to nodes, which can be thought of as a node's "height" in the graph. *Thing* nodes are always zero-order. Because edges always originate at one node and end at another node, we can call the originating node the *parent*, and the ending node the *child*, at least with respect to that edge, and define the parent's order to be one more than the maximum of the order of its children. Examples of nodes of different graph structures and the orders of the highest node are shown in Figure 13.

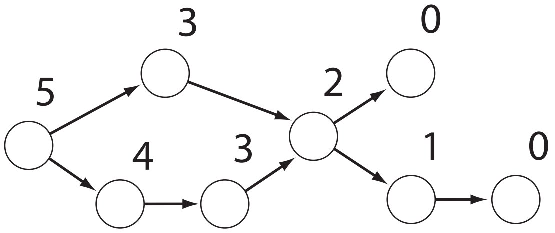


Figure 13: A graph where the orders of the nodes are numbered.

Our representation also incorporates a rough model of semantic memory called *thread memory* (Vaina & Greenblatt, 1979). This memory is implemented by attaching to each node in the episodic graph (described above) a collection of ordered lists (*threads*) of hierarchy terms to which the object or concept represented by the node belongs. In other words, each object maintains one or more sequences of class membership strings, and each such sequence is called a thread. For example, a person might be described with the following thread.

```
ambassador--diplomat--politician--human--primate--animal--thing
```

This indicates that the most specific class of which this node is a member is the `ambassador` class, and the most general class the `thing` class. We could compare this description with that of another person, say a fireman, who might be described as in the following thread.

```
fireman--rescuer--human--primate--animal--thing
```

When comparing the two, we would find that they match on the last four classes, but not the others. As a representational convenience, predicates applying to an object or relation, such as "red" or "tall", are also included on threads. By comparing threads we can obtain a rough measure of semantic similarity. This is discussed in more detail in Section A.2

There are few additional representational conventions. Our representation is intended to emphasize the metaphorical nature of human language, in that often we can cast verbs as relations between an object and a path (Lakoff & Johnson, 1980). This form of representation is also inspired by Jackendoff's descriptions of trajectories. A path in our representation is a Sequence node with any number of Derivative nodes as its elements, where the subgraphs headed by the Derivatives represent proposition phrases. An example is shown in Figure 14. We use these constructions for all motion verbs as well as those verbs that can be metaphorically related to motion. To help capture the spirit of *higher-order* nodes later in this work, we conventionally assume that path constructions are of order zero.

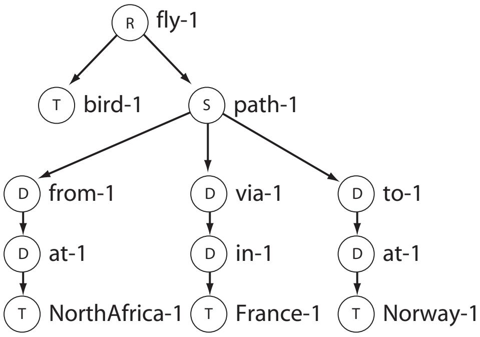


Figure 14: Representation of the sentence with paths: "The bird flew from North Africa to Norway via France."

Finally, when constructing graphs we add nodes we call *Trajectory Ladders* and *Trajectory Spaces*, both of which are Sequence nodes. Each description contains a single Trajectory Space node, a node which is not included in the node counts, nor are used for feature decomposition; rather, they serve as a handle for a description. The elements of a Trajectory Space Sequence node are Trajectory Ladder nodes, also Sequences. The Trajectory Ladders have as their elements particular actions or events. The position of these nodes in a graph description are illustrated in Figure 15. Causal and temporal relations are placed between Trajectory Ladders, not just between the elements of the ladders. The reason for this can be seen in Figure 15, where the conjunction of two events is the result of the first events.

We made use of the so-called *Karla the Hawk* dataset in Experiment 3. Doing this with our implementation required us to translate the Karla the Hawk representation into our own representation. A detailed overview of the representation used in the Karla the Hawk dataset can be found in (Gentner et al., 1993). Object nodes in the Karla the Hawk representation were straightforwardly translated to Thing nodes, but predicates—description nodes likes "red" or "tall"—were incorporated as thread elements on the appropriate Thing. Relations were straightforwardly translated, and some additional representational structure, such as Trajectory Ladder and Trajectory Spaces, were added as our representational conventions dictate. This is important because of its effect on node count comparisons—overall node counts for the Karla the Hawk representations are for the translated representations (on which our experiments were run), not the original representations.

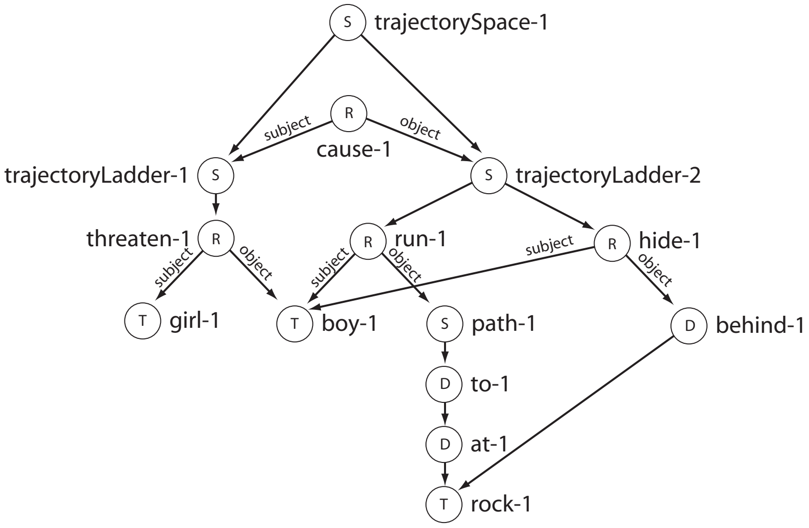


Figure 15: Representation of the scenario "The boy ran to the rock and hid because the girl threatened him."

A.2 Retrieval Algorithm

This section contains a detailed algorithmic description of how we locate features from the target in a source, and how we assign that source a retrieval score. An overview of the retrieval model can be found above in Section 2.1.

The basic outline of the algorithm is as follows: We first break both descriptions into fragments. Each feature on the target list is compared sequentially with every feature on the source list, producing a set of scores. The feature pair with the highest score is considered a match, and if the score of that match passes the scoring condition[9] those features are taken off both lists and the match score is added to the total source score. The next feature on the target list is then matched in the same manner, until all features on the source list have been matched with one feature from the target list or one of the two lists is exhausted.

A key issue in our model, and in the mechanism in general, is how we break our graph representations into features. In our current implementation, a feature is a complete sub-graph of the whole representation. The feature has a single head node and includes that node and all of its descendants. Thus, in any graph, there will be a feature corresponding to each node in the representation. As can be seen, the general size of the features runs from small (e.g., an individual object, such as "boy") to medium (such as a path), to large (such as the whole-graph-subsuming *causes* relation).

[9] Defined as falling between the upper and lower thresholds.

Algorithm 1: The main retrieval algorithm.
Retrieve(*source*, *target*)
Input: A target description, *target* (a graph), and a source description, *source* (also a graph).
Output: A real number, $score_{total}$, indicating the retrieval strength for this source relative to this target.

```
$features_{target} \leftarrow$ List-All-Unique-Nodes($target$)
$features_{source} \leftarrow$ List-All-Unique-Nodes($source$)
$score_{total} \leftarrow 0$
foreach $node_{target} \in features_{target}$
    $score_{max} \leftarrow 0$
    foreach $node_{source} \in features_{source}$
        $score \leftarrow$ Compare-Subgraph($node_{source}$, $node_{target}$)
        if $score > score_{max}$
            $node_{max} \leftarrow node_{source}$
            $score_{max} \leftarrow score$
    if Scoring-Condition($score_{max}$)
        Remove-Element($node_{max}$, $features_{source}$)
        $score_{total} \leftarrow score_{total} + score_{max}$
return $score_{total}$
```

Algorithm 2: Algorithm for comparing two features.
Compare-Subgraph($node_{target}$, $node_{source}$)
Input: Two nodes to be compared.
Output: A real number indicating how well the two nodes and all their children match.

```
if IncompatibleTypes?($node_{target}$, $node_{source}$) then return 0
$score \leftarrow$ Compare-Nodes($node_{target}$, $node_{source}$)
if No-Children?($node_{target}$) or No-Children?($node_{source}$) then return $score$
$children_{target} \leftarrow$ Get-Children($node_{target}$)
$children_{source} \leftarrow$ Get-Children($node_{source}$)
$combinations \leftarrow$ Enumerate-All-Combinations($children_{target}$, $children_{source}$)
$score_{max} \leftarrow 0$
foreach $combination \in combinations$
    $score_{combo} \leftarrow 0$
    foreach $pair \in combination$
        $childnode_{target} \leftarrow$ First-Element($pair$)
        $childnode_{source} \leftarrow$ Second-Element($pair$)
        $score_{combo} \leftarrow score_{combo} +$ Compare-Subgraph($node_{source}$, $node_{target}$)
    if $score_{combo} > score_{max}$
        $score_{max} \leftarrow score_{combo}$
return $score + score_{max}$
```

Algorithm 3: Comparison function for comparing individual nodes.

```
COMPARE-NODES($node_{target}$, $node_{source}$)
Input: Two nodes to be compared.
Output: A real number between 0 and 1 that reflects the ratio of the union of classes on
the two nodes to the intersection of class on the two nodes.
    $C_1 \leftarrow$ set of all thread elements on $n_1$
    $C_2 \leftarrow$ set of all thread elements on $n_2$
    $U \leftarrow$ UNION($C_1$, $C_2$)
    $I \leftarrow$ INTERSECTION($C_1$, $C_2$)
    if $|U| = 0$ then return 0
            else return $|I|/|U|$
```

Algorithm 4: Scoring function for Retrieval Algorithm.

```
SCORING-CONDITION($score$)
Input: A real number, $score$, and two global variables, the lower and upper thresholds,
$T_{lower}$ and $T_{upper}$.
Output: true if the score is between the thresholds, $T_L$; false otherwise.
    if $T_{lower} < score$ AND $score < T_{upper}$ then return true
                                                     else return false
```

To produce a score between two features, we first match up the root nodes of the features. If the two nodes are of the the same type (either Thing, Relation, Sequence, or Derivative), we count the number of thread elements shared between those two nodes, and then repeat the process on the nodes' children, adding the children's scores to the target, and so on recursively down the tree. If at any point two nodes do not share the same type, the count for those two nodes returns zero. All possible pairings of the children are produced by Algorithm ENUMERATE-ALL-COMBINATIONS, and the loops choose the set of children matches that maximizes the score.

By comparing threads we can obtain a rough measure of semantic similarity. We do this as shown in Algorithm 3. This algorithm divides the number of thread elements in common between two nodes by the number of unique thread elements on both nodes to give a measure of semantic similarity between 0 and 1. If two nodes share no thread elements, they will have a score of 0. If they share all their thread elements, they will get a score of 1. Examples of this calculation are given in Figure 17.

A.3 Computational Complexity

The order of our algorithm is $O(n^2)$ in the maximum number of nodes n of the target and source, which was concluded as follows.

We will proceed by first calculating the time required to precompute the results of Algorithms 2 (COMPARE-SUBGRAPH) and 3 (COMPARE-NODES). This makes the calculation of the complexity of overall algorithm easier, since the COMPARE-SUBGRAPH calls itself in a recursive fashion. By considering that the results are precomputed and cached, we can then consider the call to COMPARE-SUBGRAPH in Algorithm 1 (RETRIEVE) as a constant, and merely add the complexity of the precomputation directly to the complexity of the main RETRIEVE algorithm.

Let n be the number of nodes in the target or the source, whichever has the most nodes. To precompute COMPARE-NODES, it would have to be called n^2 times, once for each possible pair of nodes from the source and target. Let d be the maximum number of thread elements on any node. Since the union and intersection operations are linear in the number of thread elements of the nodes, i.e., $O(d)$, the contribution of Algorithm 3 is n^2d.

To precompute COMPARE-SUBGRAPH, it would also have to be called n^2 times. ENUMERATE-ALL-COMBINATIONS returns a set with all possible combinations (as opposed to permutations) of the children of the two nodes. Let k be the maximum number of children of any node found in either the target or source. Each pair of nodes has at most $k!$ sets of combinations of their children, each of which have at most k pairs of children in them, so Enumerate-All-Combinations has running time of at most $O(k!k)$. It is called once for each call of Compare-Subgraph for a contribution of $n^2k!k$. The loop at line (8) of Compare-Subgraph is called $k!$ times, and the loop at line (10) is called k times, for a total contribution of $n^2k!k$, which, when added to $n^2k!k$, does not increase the order of the complexity. Thus the total time to precompute Compare-Subgraph is on the order of $n^2k!k$.

Now we consider the main algorithm, Retrieve (Algorithm 1). Lines (1) and (2) together have contribution of n. The loops at (4) and (6) are run at most n times each, for a total contribution of this nested loop of n^2. We have considered Compare-Subgraph to have been precomputed, so it's contribution is constant. Furthermore, the scoring algorithm, Algorithms 4 is constant time. So the contribution of this topmost algorithm is $n + n^2$, if the other algorithms are considered to have been precomputed. Adding up the contributions of all the precomputations and the main algorithm, we have:

$$n + (1 + k!k + d)n^2 \tag{1}$$

We can consider k, the maximum number of children of any node, a constant term because it does not vary with n, and so drops out in the limit. We can also consider d, the maximum number of thread elements, a constant for the same reason. Thus the worst case running time of the Retrieve algorithm is $O(n^2)$ for a single source and target.

A.4 Calculating Match Scores

Because the calculation match score, done in the Compare-Subgraph algorithm using the Compare-Nodes algorithm, is the central to the operation of the model, here we give a number of examples of calculations so the reader can see how this is roughly analogous to the 'merit' or 'informativeness' of the target's feature relative to the source. In Figure 16, we show examples of comparisons of individual nodes. These scores are then used in Figure 17 to calculate match scores between features.

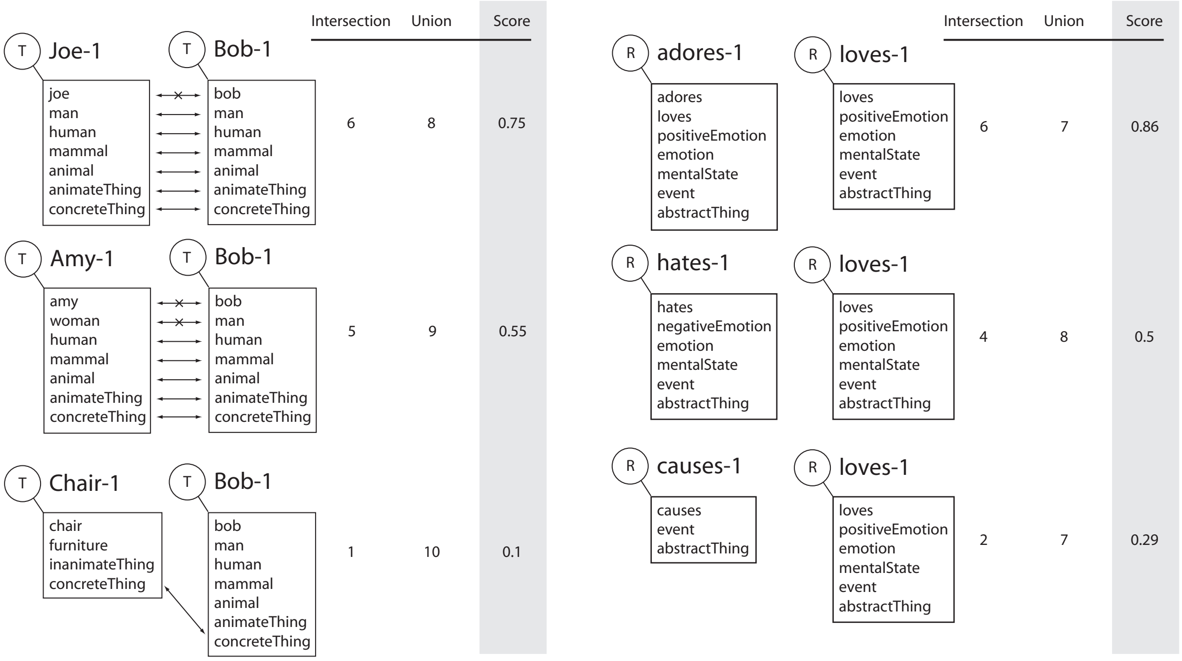


Figure 16: Examples of Node-Node Semantic Similarities as measured by the Compare-Nodes Algorithm

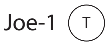
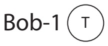
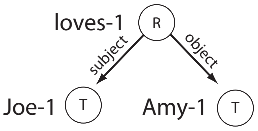
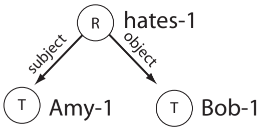
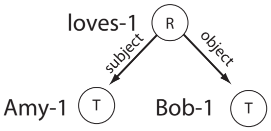
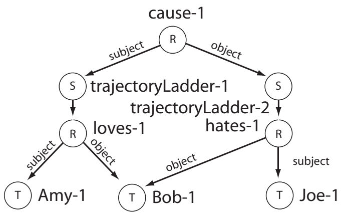
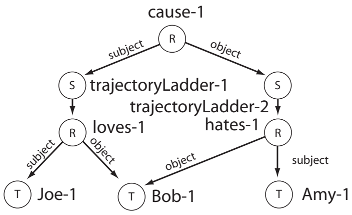
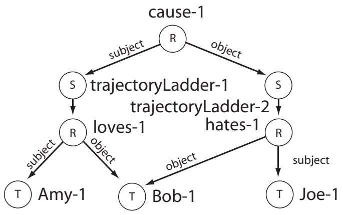
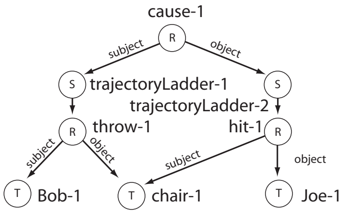
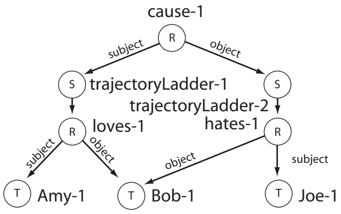
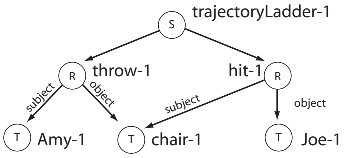
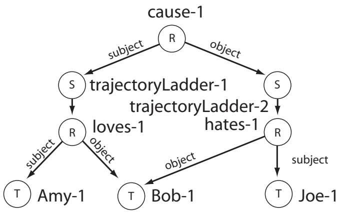
1st Feature	2nd Feature	1st Size	2nd Size	Overall Semantic Similarity	Score	Maximum Possible Score
		small	small	high	0.75	1
 loves-1 R subject object Joe-1 T Amy-1 T	 R hates-1 subject object T Amy-1 T Bob-1	small	small	medium	1.6	3
 loves-1 R subject object Amy-1 T Bob-1 T	 cause-1 R subject object S trajectoryLadder-1 S trajectoryLadder-2 R loves-1 hates-1 R subject object object subject T Amy-1 T Bob-1 T Joe-1	small	interm.	low	0.28F	3
 cause-1 R subject object S trajectoryLadder-1 S trajectoryLadder-2 R loves-1 hates-1 R subject object object subject T Joe-1 T Bob-1 T Amy-1	 cause-1 R subject object S trajectoryLadder-1 S trajectoryLadder-2 R loves-1 hates-1 R subject object object subject T Amy-1 T Bob-1 T Joe-1	interm.	interm.	high	7.72	8
 cause-1 R subject object S trajectoryLadder-1 S trajectoryLadder-2 R throw-1 hit-1 R subject object subject object T Bob-1 T chair-1 T Joe-1	 cause-1 R subject object S trajectoryLadder-1 S trajectoryLadder-2 R loves-1 hates-1 R subject object object subject T Amy-1 T Bob-1 T Joe-1	interm.	interm.	medium	4.93	8
 trajectoryLadder-1 S R throw-1 hit-1 R subject object subject object T Amy-1 T chair-1 T Joe-1	 cause-1 R subject object S trajectoryLadder-1 S trajectoryLadder-2 R loves-1 hates-1 R subject object object subject T Amy-1 T Bob-1 T Joe-1	interm.	interm.	low	0	6

Figure 17: Examples of matches between features of various sizes and semantics, and the scores they produce.