Retrieving Semantically Distant Analogies with Knowledge-Directed Spreading Activation*

Michael Wolverton and Barbara Hayes-Roth
Knowledge Systems Laboratory, Stanford University
701 Welch Rd, Bldg. C
Palo Alto, CA 94304
{mjw|bhr}@hpp.stanford.edu

Abstract

Techniques that traditionally have been useful for retrieving same-domain analogies from small single-use knowledge bases, such as spreading activation and indexing on selected features, are inadequate for retrieving cross-domain analogies from large multi-use knowledge bases. In this paper, we describe Knowledge-Directed Spreading Activation (KDSA), a new method for retrieving analogies in a large semantic network. KDSA uses task-specific knowledge to guide a spreading activation search to a case or concept in memory that meets a desired similarity condition. Specifically, KDSA exploits evaluations of near-analogies encountered during the search to direct the search toward progressively more promising analogies. We describe a specific instantiation of this method for the task of innovative design, and we summarize the theoretical and experimental results used to validate KDSA.

Introduction

Cross-domain analogy is a commonly-used reasoning device, especially among individuals who must exhibit a high level of creativity in their reasoning. Authors, journalists, and political speechwriters often use surprising metaphors in order to enliven their prose; clever teachers use analogies to familiar concepts outside of the domain of discussion to explain unfamiliar concepts; and engineers and inventors often use analogies in order to help them produce a novel design. The concern of this paper is the retrieval and use of cross-domain analogies, specifically those in which the two analogues are *semantically distant* from one another—that is, they are very different from one another in all but a few key features.

The literature on invention is full of examples of inventions that were guided by semantically distant analogies. Gutenberg invented the printing press after noticing the connection between applying force to impress script on paper and applying force to squeeze grapes in a wine press (Koestler 1965) . Edison's invention of the quadruplex telegraph was based almost entirely on an analogy to a water system of pumps, pipes, valves, and water wheels (Hughes 1971). And actress Hedy Lamarr conceived of a method for coordinating frequencies between sender and receiver in frequency-hopping communication by analogy to a player-piano roll (Simon *et al.* 1985). These examples all show the inventor making a connection between two concepts not normally thought of as connected. The fact that human inventors use semantically distant analogies in their reasoning suggests that these analogies can be an important technique for computer reasoning as well.

From looking at examples of analogies in invention, we can surmise two characteristics of semantically distant analogies that present special problems for the development of a computational model:

(1) The domains from which the analogies are drawn are unpredictable. The concepts used to guide novel designs come from a wide range of domains, and it is impossible to predict, given the target design domain, which base domain(s) may prove fruitful for drawing useful analogies.

(2) In the analogies that are made, differences between the analogous concepts are as important as similarities. An inventor's chances of developing a truly novel design by analogy are greatly increased by using a base concept that is unusual or unexpected. This suggests that the base concept used should be as different as possible from the target concept while still being useful for design. That is, the two concepts should share only those features that are necessary to the function of the invention, *and should mismatch on as many extraneous features as possible.* In particular, analogies with a high degree of surface similarity seem unlikely to be useful in producing novel inventions.

These two characteristics provide reasons that existing approaches to analogy retrieval are inappropriate for retrieving semantically distant analogies. Most existing approaches to analogy retrieval are based either on task-specific indexing of concepts in a case library or on spreading activation in a semantic network, but

*This research was supported by NASA Grant NAG2-581, by Texas Instruments Contract 7554900, and by McDonnell Douglas Contract S07705. Thanks to Rich Washington and the anonymous AAAI reviewers for helpful comments.

neither of these general approaches is well-suited for finding semantically distant analogies. The indexing approach is inappropriate because characteristic (1) above suggests that a successful "case library" for semantically distant analogies would in fact be a large multi-domain multi-use knowledge base, but most successful indices in case-based reasoning are task-specific. To create a new set of indices for each possible task that may be performed in such a KB (and each possible analogical use of a given concept) would require a prohibitive number of organizational links or constructs. The spreading activation approach is inappropriate because characteristic (2) above suggests that most corresponding features involved in the analogues will be far from each other in the semantic network, and an uncontrolled spread of activation throughout the large semantic net will bog down in combinatorial explosion before reaching the semantically distant base concepts it seeks.

This paper introduces a method, called *knowledge-directed spreading activation* (KDSA), for retrieving semantically distant analogous concepts from a large diverse knowledge base. This method is based on controlled search in a general semantic network. It uses task-specific knowledge to guide a series of spreading activation searches from the target concept to a semantically distant base concept. This knowledge is applied in the evaluation of intermediate concepts retrieved by a standard spread of activation, and by the modification of weights controlling the spread of activation based on those evaluations. The next section describes this method in more detail.

The Approach—KDSA

Viewed abstractly, KDSA is an application of general techniques from state-space search (evaluation functions, subgoaling, etc.) to knowledge base search. KDSA finds analogues by a *series* of heuristically-guided spreading activation searches. Each time spreading activation retrieves a concept from the knowledge base, the concept is evaluated as an analogue, and that evaluation is used to direct the next spreading activation search in more promising directions. KDSA uses promising concepts retrieved during these spreading activation searches as "beacons", guiding the search successively closer to a semantically distant base.

This description of KDSA will assume that all world knowledge is represented in a single semantic network. Within that semantic network, small subgraphs of nodes and links which represent aggregate concepts are explicitly grouped together as conceptual graphs (Sowa 1984). Individual conceptual graphs are treated the same as primitive nodes—i.e., they can be associated with other nodes via links, and they can themselves be parts of larger conceptual graphs. In the discussion below, conceptual graphs will be referred to merely as "concepts".

1. Assign activation to all nodes in the target.
2. Spread activation in semantic network until a new intermediate concept (IC) is retrieved.
3. (GRAPH MATCHER) Find the best mapping between target and IC based only on maximizing isomorphism and minimizing semantic distance between nodes.
4. (MATCH EVALUATION) Evaluate the mapping according to domain-specific similarity metric. If evaluation meets the metric, return IC as base and exit.
5. (SEARCH CONTROL) Based on evaluation, alter the state of the semantic network to guide the next phase of spreading activation in a more promising direction.
6. Go to 2.

Figure 1: Knowledge-Directed Spreading Activation

The basic algorithm of KDSA is shown in Figure 1. The low-level search of memory (step 2 in the figure) is conducted by a spreading activation mechanism (see, e.g., (Anderson 1983)). In this formalism, activation is passed from node to adjacent node via the links that connect them until one concept accumulates enough aggregate activation to be considered retrieved. This basic spreading activation model is a blind knowledge search mechanism. Some method of controlling the search is necessary for the system to retrieve the types of semantically distant base concepts described in the introduction. KDSA uses feedback from the analogues retrieved so far to focus the search.

The agent architecture encompassing KDSA begins the retrieval process when some executing task requests an analogy and designates a target concept. This initial request causes some nodes in the semantic network—those representing the target concept plus possibly others representing desired features of the solution, etc.—to be assigned activation, and this assignment begins the spread of activation in memory. When a concept is retrieved by the spread of activation, the *graph matcher* computes a mapping between it and the target concept. The *match evaluation component* then forms an evaluation of the mapping based on a task-specific similarity metric. This evaluation is passed on to the *search control component*, which uses its task-specific heuristics to focus the spreading activation search in directions that are more likely to lead to highly-evaluated analogies for the current task. The process repeats until an analogue that meets the matching component's similarity metric is retrieved.

For simplicity, KDSA has been described so far as a strictly serial algorithm. In fact, it is designed (and implemented) as a collection of independent knowledge sources that execute within a larger intelligent agent architecture[1], and that interact with the agent's other

[1]In the computer implementation of KDSA, the agent architecture used was BB1 (Hayes-Roth 1990).

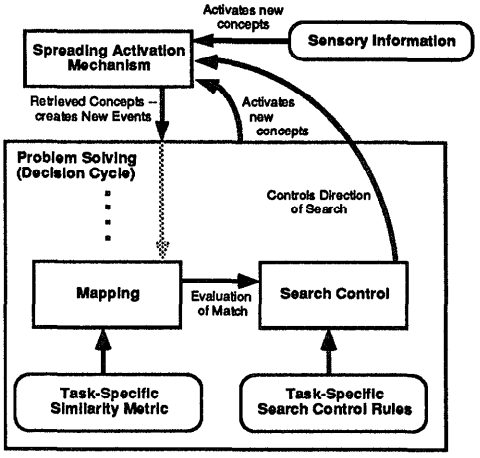


Figure 2: Integration of KDSA into problem-solving architecture

activities. Figure 2 shows this interaction. At any time during the cycle of Figure 1, other concepts may be activated by the agent's other activities, such as ordinary problem solving or processing sensory input. In this way KDSA can account for an individual possibly "stumbling across a solution", i.e., being reminded of an analogue by external or internal cues.

The important components of the retrieval system are discussed in more detail below.

Match Evaluation Each time a concept is retrieved by the spreading activation search as a potential base concept, it is passed to the matching component. The matching component first forms the best possible partial mapping between the potential base and the target, and then it evaluates that partial mapping using heuristics that are specific to the task for which the analogy will be used. These heuristics will base their evaluation on three features of the partial mapping: (1) semantic distance between corresponding nodes in the mapping, i.e., the minimum path distance in the type hierarchy between corresponding nodes of the mapping (2) isomorphism between the graphs, i.e., how many nodes and links match between the target and potential base, and (3) the portion of the representation of the target concept matched, and the relevance of that portion to the goal. The evaluation consists of a numeric rating of the mapping, and a description of the shortcoming(s) of the mapping assigned by the heuristics. If the numeric rating is greater than a threshold value, the potential base is accepted as the final analogy, and the KDSA process halts. Otherwise, the evaluation is passed on to the search control component.

Search Control The search control component uses evaluations from the match evaluation component and other information about the state of the search to influence the direction of the spread of activation. It uses heuristics to control the direction of the search in two ways: (1) it can change activation of concepts in the semantic net, particularly the target concept and the retrieved intermediate concept, and (2) it can modify the condition under which spreading activation will retrieve new intermediate concepts. The first of these, changing activation of selected concepts in the KB, is the more important of the two methods of search control. This method includes strengthening the activation of promising intermediate concepts (those that nearly pass the mapping component's similarity metric for being a good final analogy), weakening the activation of unpromising concepts, changing activation of *portions* of the intermediate concept or the target based on evaluations, and clearing the activation of all nodes in the semantic network (to start the search over from a new state).

A simple use of KDSA's search control would have it clearing all activation in the semantic network each time a promising concept is encountered, and then restarting the search by making the promising concept a source of activation. In this way KDSA can use these promising concepts as *beacons* along the way to the final good analogy. This is very similar to the way that promising intermediate states are used in heuristic search techniques such as hill-climbing or best-first search (Pearl & Korf 1987).

The use of the matching component of the mechanism to provide feedback to the spreading activation search provides a key distinguishing feature of our approach. Most previous approaches to analogy serialize the retrieval and mapping processes: first they retrieve a concept, then they try to map it, then if mapping fails they start at ground zero with retrieval again. By contrast, mapping in KDSA is an integral part of retrieval: mapping (the matching component) provides ongoing information to the retrieval mechanism (spreading activation and search control) throughout the duration of the retrieval process.

KDSA Applied to Innovative Design

This section describes the particular heuristics used in the implementation of KDSA, called IDA (for Innovative Design by Analogy), to find analogies that are useful for guiding an innovative redesign of the target.

IDA operates in a knowledge base of devices, natural or man-made systems that perform some function. The knowledge base may contain definitions of other concepts as well, but IDA requires that each device be represented by its structure, behavior, and function. Representations of structure consist of the device's parts along with different types of connections among those parts. Representations of behavior and function consist of chains of primitive processes along with the individuals (structural components, substances, etc.) on which those processes act. IDA takes as input an existing device, and returns as output an abstract re-

design of that device which satisfies the device's top-level functional requirements, but does so in a different way. This redesign consists of a replacement of one of the target device's top-level behaviors with a behavior from the base device. E.g., a behavior like SPRAYING from the representation of the sprinkler irrigation system may be replaced with DIFFUSION from the circulatory system.

The particular heuristics used in IDA's mapping component attempt to find analogues that satisfy two general requirements:

(1) The base and target devices must have similar functions, but different behaviors and structures.

(2) The base must be adaptable with regard to the target device. That is, IDA must be able to analogically adapt the retrieved base into a new device that satisfies the same function as the target device.

The purpose of the first requirement is to find an analogue which will lead to a redesign which is useful ("similar function") and at the same time novel ("different behavior and structure"). The purpose of the second requirement is to ensure that IDA will actually be able to produce a redesign based on the retrieved base concept, i.e., that the mismatch in behavior with the target is not so great that the two devices have nothing to do with one another. Thus this second requirement's implementation will depend on the system's mechanism for adapting the retrieved base into a final design.

To implement these two requirements, IDA's mapping component considers separate portions of a device's representation separately. Each device representation is broken down into structure, behavior, and function. The behavior and function representations are broken down further into (1) a sequence of primitive processes that make up the behavior or function, and (2) the individuals on which those processes act. There are separate requirements on the degree of isomorphism and semantic distance required for each of those portions of the representation. For example, IDA prefers the match between nodes in the structures of the target and base devices to be high in semantic distance (to satisfy the dissimilar structure requirement), and prefers a mismatch on only one primitive process in the behaviors of the target and base devices (to satisfy the adaptability requirement).

After the mapping component evaluates devices according to the two requirements, the search control module must focus the spread of activation toward other devices in the KB which meet those requirements. IDA does this by focusing the search based on the strengths of the retrieved beacons encountered so far in the search. The mapping component identifies an intermediate concept as promising if it comes close to meeting the metric for being a final analogy. For each promising concept, the search control component then strengthens the activation of its portions that did meet the mapping component's individual requirement. The rest of the activation in the semantic network is wiped out, and the search is restarted from this new state.

IDA's search control rules also use abstractions in the knowledge base as "bridges" to other domains. When IDA retrieves a concept that is in the same domain as the target and is a directly-linked example of a *generic abstraction*—a concept that abstractly describes specific concepts from a number of different domains—it strengthens the activation of that abstraction. This will allow activation to be spread into other domains, increasing the likelihood that IDA will find a distant analogy.

Example

This section presents an example demonstrating the execution of KDSA to retrieve an analogy for creative design. The example shows IDA's behavior for the goal of redesigning a blinkered railroad crossing, that is, an intersection of road and railroad tracks where a train's presence on the tracks is indicated only by blinking lights signalling drivers on the road to stop. IDA meets this goal by suggesting redesign by analogy to an on-off valve. Specifically, it suggests replacing the FLASHING behavior in the description of the blinkered railroad crossing with the BLOCKAGE behavior in the description of the on-off valve. This abstract analogical specification might suggest to a human designer a railroad crossing with a gate that blocks traffic from crossing the tracks. The retrieval of the on-off valve takes place in the following steps:

1. The nodes contained in the representation of BLINKERED-RR-CROSSING are made sources of activation (i.e., they are tagged with some number), and IDA begins spreading activation.

2. After a few cycles of spreading activation, the device INTERSTATE-HIGHWAY-SYSTEM is retrieved. This device is mapped to BLINKERED-RR-CROSSING, and the mapping is evaluated. The mapping is found to be unpromising—the structures of the two devices are semantically close, and the behaviors and functions of the two devices do not correspond in any respect. However, IDA notices that INTERSTATE-HIGHWAY-SYSTEM is an instance of a generic abstraction, the FLOW-SYSTEM device. IDA recognizes this abstraction as a possible mechanism for moving the search out of its current domain, and makes FLOW-SYSTEM a source of activation. All other activation in the semantic network (except the target's) is cleared, and spreading activation starts again.

3. Another of FLOW-SYSTEM's instances, PLUMBING-SYSTEM, is retrieved next, and the mapping between it and BLINKERED-RR-CROSSING is evaluated. This mapping shows high semantic distance between the structures of the devices, and

poor matches between the behaviors and functions of the devices. IDA wants high semantic distance in structure, so the structural aspect of the mapping is rated high, but the behavioral and functional aspects of the mapping are rated low. Since the structure of the PLUMBING-SYSTEM is the strongest part of the mapping evaluation, the search control component makes PLUMBING-SYSTEM's structure a source of activation. Since the behavior and function of the PLUMBING-SYSTEM were rated low, the search control component still bases the search on the behavior and function of the target. So the structure of PLUMBING-SYSTEM and the behavior and function of BLINKERED-RR-CROSSING are made sources of activation, and all other activation in the network is cleared.

4. The next concept retrieved is ON-OFF-VALVE. The matching component recognizes that the mapping between ON-OFF-VALVE and BLINKERED-RR-CROSSING is high in semantic distance between the structures, high in isomorphism between the functions, very low in semantic distance between the top-level process sequences of the functions (they both toggle between PREVENTing and ALLOWing another process), and mismatches in a single process in the behavior description (the BLINKING of the rr crossing corresponds to the BLOCKAGE of the valve). With these conditions met, ON-OFF-VALVE meets the similarity metric for being a final analogy for innovative design. It is retrieved, and IDA's simple design module suggests replacing redesigning the BLINKERED-RR-CROSSING by replacing its BLINKING process with ON-OFF-VALVE's BLOCKAGE process.

Results

One of the major questions important in the evaluation of KDSA is: will KDSA retrieve analogies without examining a sizable fraction of the entire knowledge base? In order to answer this question, KDSA was evaluated using two complementary methods.

The first of these methods is to analyze the behavior of a theoretical model. This model predicts KDSA's retrieval time given various parameters such as the size of the knowledge base, the semantic distance required for the analogy, the likelihood of encountering a beacon concept in the knowledge base, and the quality of each beacon concept in terms of the benefit it provides in reaching the ultimate base concept. This model allows us to examine the behavior of KDSA under a wide range of problem and knowledge base characteristics.

The second method of evaluating KDSA is to examine the behavior of the implementation, IDA. This implementation of KDSA demonstrates that KDSA can, in fact, automatically retrieve semantically distant analogies which are useful in solving a real problem. In addition, while it is presently impossible to test IDA with an actual very large knowledge base, we can measure IDA's retrieval time as a function of the KB size for various subsets of IDA's small knowledge base. These experiments allow us to examine KDSA's behavior as the knowledge base grows, and compare that actual behavior to the prediction of the theoretical model.

Figure 3 graphs some of the results produced by these two validation methods. It shows time taken to retrieve a semantically distant analogy as the size of the knowledge base grows, both for (a) the actual implementation operating in relatively small knowledge bases, and (b) the theoretical model as the knowledge base grows to a size of 1 million nodes. Each graph also shows retrieval time for standard spreading activation (SA) as well. Both methods showed retrieval time for KDSA growing much more slowly than for standard spreading activation as KB size grows. The theoretical model predicts behavior that is roughly logarithmic in the size of the KB.

Detailed presentations of the theoretical and experimental results are published in (Wolverton 1994). These results can be summarized with the following four qualitative statements, with the first statement being verified by both the theoretical model and experiments, and the remainder being predicted by the theoretical model:

(1) As the knowledge base size grows, retrieval time with KDSA grows much more slowly than does retrieval time with standard SA.

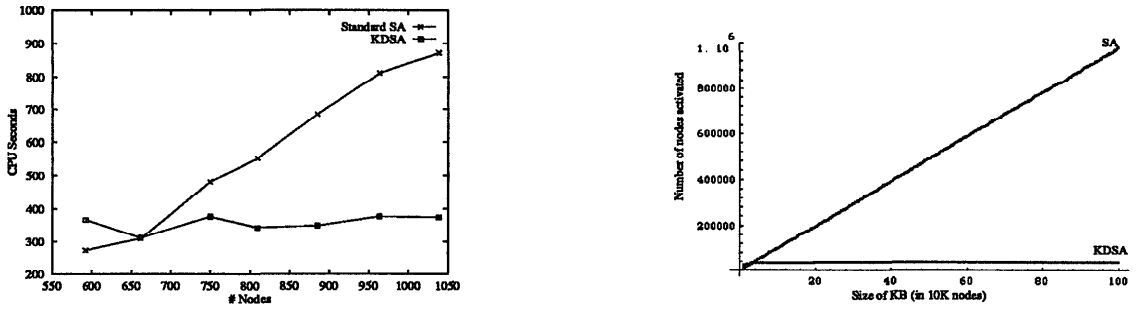
(2) For analogies in which the target and the base are semantically distant, KDSA is far more efficient than standard SA.

(3) KDSA is robust over different distributions and utilities of beacon concepts in the knowledge base. Even when the benefit of each beacon search is low relative to the effort involved, KDSA still shows significant savings over standard SA.

(4) KDSA is robust in the face of bad beacons. When a KDSA search suffers from beacons that direct the search away from, rather than toward, the eventual base, KDSA still shows substantial savings over standard SA.

Related Work

There is a large body of AI literature on information retrieval in semantic networks. SCISOR (Rau 1987) and GRANT (Cohen & Kjeldsen 1987) both use heuristic information to direct a spread of activation in semantic networks. KDSA's search control component is similar to the relatedness condition which controls the spread of markers in SCISOR and the path endorsements which direct spreading activation in GRANT. KDSA differs from these systems, however, in that it uses information from previous match evaluations to dynamically adjust the direction of the spread of activation. KDSA in effect runs a *series* of SCISOR-like or GRANT-like searches, starting each sub-search



(a) (b)

Figure 3: Retrieval time for KDSA and standard SA as KB size grows, (a) as observed in the computer implementation IDA in a small knowledge base, (b) as predicted by the theoretical model in a large knowledge base

from the near-misses it has encountered in previous sub-searches, and using the evaluations of those near-misses to formulate its search control for the next sub-search.

Other researchers have used spreading activation in semantic networks to retrieve analogues. Holland et. al.'s PI (Holland *et al.* 1986), Anderson's PUPS (Anderson & Thompson 1989), and Jones's EUREKA (Jones 1989) are all general cognitive models which use spreading activation for analogue retrieval (as well as other knowledge retrieval). In all of these approaches, the architecture's ability to control the spread of activation is limited, so they will have difficulty retrieving semantically distant analogies without the help of external cues. KDSA is able to use such cues when they are available, but also is able to retrieve semantically distant analogies spontaneously.

Conclusion

We have presented knowledge-directed spreading activation, a task-independent method for retrieving analogues in a multi-domain knowledge base. KDSA overcomes the shortcomings that indexing methods may have in large knowledge bases by applying task-specific knowledge to a general semantic network search technique. And it extends previous models of analogue retrieval by using evaluations of failed analogies encountered in the earlier stages of the search to influence the direction of the search in later stages. KDSA has been shown to be useful in retrieving analogues for a real task in the computer implementation IDA. And experimental and theoretical results provide evidence that KDSA will be tractable in a large knowledge base.

References

Anderson, J. R., and Thompson, R. 1989. Use of analogy in a production system architecture. In Vosniadou, S., and Ortony, A., eds., *Similarity and Analogical Reasoning.* Cambridge: Cambridge University Press. 267–297.

Anderson, J. R. 1983. *The Architecture of Cognition.* Harvard University Press.

Cohen, P. R., and Kjeldsen, R. 1987. Information retrieval by constrained spreading activation in semantic networks. *Information Processing and Management* 23(4):255–268.

Hayes-Roth, B. 1990. Architectural foundations for real-time performance in intelligent agents. *Journal of Real-Time Systems* 2:99–125.

Holland, J. H.; Holyoak, K. J.; Nisbett, R. E.; and Thagard, P. R. 1986. *Induction: Processes of Inference, learning, and Discovery.* Cambridge, Massachusetts: MIT Press.

Hughes, T. P. 1971. How did the heroic inventors do it? *American Heritage of Invention and Technology* 1(2):22–23.

Jones, R. 1989. Learning to retrieve useful information for problem solving. In *Proceedings of the Sixth International Workshop on Machine Learning,* 212–214.

Koestler, A. 1965. *The Act of Creation.* Macmillan.

Pearl, J., and Korf, R. E. 1987. Search techniques. *Annual Review of Computer Science* 2:451–467.

Rau, L. F. 1987. Knowledge organization and access in a conceptual information system. *Information Processing and Management* 23(4):269–283.

Simon, M. K.; Omura; Scholtz; and Levitt. 1985. *Spread Spectrum Communications, Vol. 1.* Computer Science Press.

Sowa, J. F. 1984. *Conceptual Structures: Information Processing in Mind and Machine.* Addison-Wesley.

Wolverton, M. 1994. *Retrieving Semantically Distant Analogies.* Ph.D. Dissertation, Computer Science Department, Stanford University.