SIMILARITY NETWORKS AS A MEANS OF INDEXING AND RETRIEVING DESCRIPTIONS

by

David Lee Bailey

SUBMITTED TO THE DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING IN
PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1986

Copyright (c) 1986 Massachusetts Institute of Technology

Signature of Author
Department of Computer Science and Engineering
June 2, 1986

Certified by
Professor Patrick H. Winston
Thesis Supervisor

Accepted by
Professor David Adler
Chairman, Department Committee
SIMILARITY NETWORKS AS A MEANS OF INDEXING AND RETRIEVING DESCRIPTIONS

by

David Lee Bailey

Submitted to the Department of Computer Science and Engineering on June 2, 1986 in partial fulfillment of the requirements for the degree of Bachelor of Science.

Abstract

Memory systems need to provide an efficient means of retrieving stored items. Similarity networks may provide a means for achieving efficient retrieval. A similarity network is formed by linking the descriptions of objects in it that are deemed similar enough by some model of similarity. The best methods of constructing, searching, and representing a similarity network are studied to determine the effectiveness of using similarity networks as a means of indexing and retrieving descriptions of objects.
Dedication

To Professor Winston for his guidance and encouragement, to Carrol for her support and understanding, and to my parents for putting me through MIT.
Table of Contents

Abstract 2
Dedication 3
Table of Contents 4
List of Figures 5
List of Tables 6

1. Introduction 7

2. The Construction of Similarity Networks 10
   2.1 Background 10
   2.2 Experiments 13
   2.3 Results 16

3. Searching a Similarity Network 20
   3.1 Background 20
   3.2 Experiments 21
   3.3 Results 21

4. Representing a Similarity Network 24
   4.1 Background 24
   4.2 Experiments 26
   4.3 Results 27

5. Future Directions 29

6. Conclusions 31

Appendix A. The Description of an Armchair 33

Appendix B. The Similarity Network Produced for the Furniture Domain 34

Appendix C. A Connectivity Histogram for Furniture 36

Appendix D. Local Connectivity for Furniture 37

Appendix E. The Representation of the Furniture Network 38
List of Figures

Figure 1-1: An Example of a Similarity Network 9
Figure 1-2: Searching a Similarity Network 9
Figure 2-1: The Changing Connectivity of a Network 14
Figure 2-2: Depth of Connectedness Example Piece 16
Figure 4-1: The Searches Developed 26
List of Tables

Table 2-I: Range of Thresholds of Connectivity 17
Table 2-II: Resulting Thresholds from Average Size 18
Table 2-III: Resulting Thresholds from Average Depth 19
Table 3-I: Results of Searching with a Memory 22
Table 3-II: Results of Searching by Match Scores 22
Table 4-I: Results of Searching with Sorted Pieces 27
Chapter 1

Introduction

An important part of any memory system is the ability to retrieve the items that are stored in the system quickly and efficiently. One possible method of achieving efficient retrieval involves using similarity networks as an encoding and storage mechanism.

A similarity network is a means of storing descriptions of objects according to how similar the objects are to each other. A similarity network is formed by matching the attributes of objects and linking those objects whose similarity score is greater than a given threshold. Two areas in which similarity objects may be of use are in learning and reasoning systems. For example, a similarity network might be used to index the Shakespeare stories and examples in Winston's learning by precedents system [Winston 81]. Another system which might benefit from the use of similarity networks is the Mechanics Mate [Brady 84], which needs to store descriptions of tools and fasteners.

Similarity networks were first described in Winston's thesis "Learning Structural Descriptions from Examples" [Winston 75]. They are later mentioned in Minsky's "A Framework for Representing Knowledge" [Winston 75], but no attempts were made to actually implement similarity networks until now.

In my research I studied three aspects of similarity networks in order to determine their efficiency and usefulness as a storage mechanism. The three questions to be answered were as follows:

1.) What is the best way to construct a similarity network?
2.) What is the most efficient method of
searching a similarity network?

3.) How can a similarity network be represented to allow for multiple networks to be linked and searched?

During the course of my research, I found a way to construct similarity networks automatically from the descriptions of the objects. Figure 1-1 illustrates the furniture network that was constructed using this procedure.

I have also developed a program to search a similarity network efficiently. Figure 1-2 shows an example of a search of the furniture network from nightstand to counter. I have devised a representation for similarity networks that can be determined by my programs while they construct the network, and I have developed a more efficient search based upon this representation.

Finally, a search has been written which uses the representations of similarity networks to search among multiple networks.
Figure 1-1: An Example of a Similarity Network

Figure 1-2: Searching a Similarity Network
Chapter 2

The Construction of Similarity Networks

2.1 Background

There are two basic considerations in the construction of a similarity network. First is the model of similarity to be used. The second is the method with which to determine the threshold for similarity. Both the proper model and the correct threshold are necessary in order to form a network that looks right and that maximizes similarity information.

Three domains were developed for experimentation: furniture, transportation, and food. The three domains vary in size from 17 members for furniture to 47 members for food. These particular domains were chosen primarily because they all consist of basic level categories such as chair, car, and soda [Mervis 81]. Basic level categories are useful for several reasons. First, they have several attributes or features in their descriptions. Second, the basic level categories are where the information value is maximized [Rosch 78]. This implies that matching features of these categories will show many differences as well as similarities. Finally, basic level categories are at a good conceptual level, meaning that they are simple and neither too general nor too specific.

The features used to describe the elements of the furniture, transportation, and food domains were of two types, functional and physical. The functional features [Winston 83] describe how an object is used, or what its purpose is. For example, the functional description of an armchair is as follows:

1.) You can sit in it.
2.) You can rest your back on it.
3.) You can rest your arms on it.

Functional descriptions are useful for describing natural concepts that have few sufficient conditions for identification. Functionality abstracts away from how an object is made or what it looks like, and focuses attention on important features. For some objects, however, it is useful to include physical descriptions or features [Smith 81] as well. Physical descriptions help distinguish between objects with the same functional description. For example, in the food domain it is helpful to describe milk as containing minerals, as well as being a beverage, to help distinguish it from water. Both the functional and the physical features are used in determining how similar or dissimilar two objects are.

Matching is done using Winston’s matcher to compare each item in a domain to all of the other items in that domain. The matcher looks at the features of the two objects that it is comparing and returns three numbers. The first number tells how many features were found in object1 that were not found in object2 (object1 \object2.) The second number tells how many features were common to both objects (object1 \\object2.) The third number tells how many features contained in object2 were not found in object1 (object2 \object1.) Using these three numbers, different models of similarity were used to determine whether or not object1 was similar to object2.

Four different models of similarity were tested using the results of the matcher on the three domains mentioned above. The first model, called the common model, simply used the second number from the matcher (object1 \\object2) for a similarity score. The idea behind the common model is that objects are similar if they have a certain number of features in common. The second model of similarity, the distinct model, used the sum of the other two numbers from the matcher, (object1 \object2) + (object2 \object1), as a similarity score. This score
was then tested to be below a threshold before similarity was determined. (The distinct model differs from the other similarity models in this respect, because all of the other models must have scores above a given threshold.) The idea here is that two objects are similar if they are not too dissimilar, that is if they do not have too many distinct features. The ratio model was used as the third model of similarity. The ratio model used all three numbers returned by the matcher, finding the ratio between the common features and all of the features, \( (\text{object1} \cap \text{object2}) / (\text{object1} \cup \text{object2}) \). The ratio model takes into account both common and distinct features when determining similarity, unlike the previous two models. The fourth model, the contrast model proposed by Tversky [Tversky 77], also uses both common and distinct features to find similarity score. The contrast model takes the common features, weights them by a constant multiplier, and then subtracts away the distinct features (also weighted by constants.) The contrast model, then, is as follows:

\[
\text{Sim}(01, 02) = -\alpha f(01-02) + \theta f(01 \cap 02) - \beta f(02-01)
\]

where \( \alpha, \beta, \) and \( \theta \) are weights, and \( f \) is used for this purpose to give the cardinality of each set. Note that both the common and the distinct models are special cases of the contrast model with either \( \alpha \) and \( \beta \) or \( \theta \) set to 0. The similarity models each return a score which is compared to a threshold to determine similarity.

The threshold at which sufficient similarity is deemed to exist is an important aspect of the construction of a similarity network. If the score returned by the similarity model meets or (in most models) exceeds the threshold, then the two objects are linked together in the similarity network. Otherwise the objects are left unlinked. The threshold provides a means of discrimination as to what scores actually constitute a similarity between objects. If the threshold is set too low,
everything in the network is linked together and the efficiency of a similarity network is lost. If, on the other hand, the threshold is set too high, only objects that are nearly identical are linked, and almost all of the similarity information is lost. At the proper threshold, however, objects cluster into pieces in the network, providing optimal similarity information about the domain.

2.2 Experiments

Experiments were designed to choose the best similarity model together with the best method of deciding on a threshold for the construction of a similarity network. Two experiments were run for the model of similarity, and one for the choice of threshold.

The quality of the similarity model was determined by two statistics. First was the connectivity (or number of pieces) of the network as the threshold was varied (see figure 2-1). This was used to discover over what range of thresholds the network would change from entirely connected (one piece) to completely broken up (N pieces, where N = the number of descriptions in the network.) A short range of thresholds would indicate that the network breaks up very quickly, leaving little choice as to which threshold to use to produce the best network. A long range of thresholds would give greater flexibility in the choice of which threshold to use, and thus allow for a more suitable network to be chosen. To study connectivity, a histogram of the number of pieces in the network over different thresholds was produced for each model of similarity over all three domains.

The second measure that was obtained was the local connectivity of each item in the network. Local connectivity chooses the best N matches (where N is an integer) in the network for each item and lists them. In this way it can be seen
Figure 2-1: The Changing Connectivity of a Network

which model of similarity produces the best matches for each object. Models of
similarity were compared over each domain for \( N = 5, 3, \) and 1. Matches were judged subjectively based upon how similar descriptions of the objects were and by how well the matches corresponded to the way that the objects are classified in the real world.

Two methods for determining the proper threshold for the construction of a similarity network were tested by this next experiment. Both centered around the idea that the best way to judge the threshold was by the pieces in the network that it produced. Good pieces are those that contain descriptions that are very similar, but no descriptions that are somewhat dissimilar to the others.

The first method for choosing a threshold was by the average size or population of the pieces in a network. This method maintains that the way to determine whether or not pieces are good is to see how many descriptions are contained in each of them.

The second method for finding the proper threshold was by checking how closely connected each description in the piece was to the other descriptions. This was measured by finding the average depth of connectedness of each member of a piece. This idea is perhaps best illustrated by an example. Take a piece with four members, A, B, C, and D, where A is connected to B and C, and B and C are connected to D (see figure 2-2). Then A is connected to B and C with a depth of one, and A is connected to D with a depth of two. The average depth of connectedness for A in this piece is thus 1.3. The depth of connectedness for B, C, and D must now be found and averaged with that for A to determine the average depth for this piece. This in turn is averaged with the depths from all of the other pieces in the network to determine the average depth of connectedness for the network at this particular threshold. From this it can be seen that the lower the average depth of connectedness, the more similar (tightly coupled) the descriptions in that piece are.
Figure 2-2: Depth of Connectedness Example Piece

Both methods for determining the proper threshold were tested in the same way. Resulting thresholds were compared to "ideal" thresholds for each network, which were chosen subjectively prior to the experiment based upon the pieces that they produced. The method producing thresholds nearest to the ideal thresholds in each case was deemed to be the proper method to use when constructing a similarity network.

2.3 Results

The results of the first two experiments narrowed the number of viable models of similarity, but did not conclusively point to any one. The experiment to determine the proper method for determining the threshold was also surprisingly inconclusive.
The connectivity histograms of the networks showed that the four models of similarity break up in a similar manner, but over quite different ranges. In general, once a network begins to break up, it continues to do so at nearly a constant rate. The exception to this is the contrast model with theta > 1, which tends to begin breaking up slowly, and disconnects more suddenly at the end. The average ranges of thresholds (see table 2-I) are quite different for each model. The common and distinct models both have short ranges, and thus break up quickly. The ratio model also appears to have a short range, but this is in part due to my choice of thresholds.

**RANGE OF THRESHOLDS**

<table>
<thead>
<tr>
<th>MODELS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMON</td>
<td>7.3</td>
</tr>
<tr>
<td>DISTINCT</td>
<td>10.7</td>
</tr>
<tr>
<td>RATIO</td>
<td>10.3</td>
</tr>
<tr>
<td>CONTRAST (ALPHA = 1, THETA = 1)</td>
<td>11.3</td>
</tr>
<tr>
<td>CONTRAST (ALPHA = 1, THETA = 2)</td>
<td>14.3</td>
</tr>
<tr>
<td>CONTRAST (ALPHA = 1.5, THETA = 2)</td>
<td>15.3</td>
</tr>
<tr>
<td>CONTRAST (ALPHA = 1, THETA = 3)</td>
<td>19.0</td>
</tr>
<tr>
<td>CONTRAST (ALPHA = 1, THETA = 4)</td>
<td>24.0</td>
</tr>
<tr>
<td>CONTRAST (ALPHA = 2, THETA = 4)</td>
<td>24.3</td>
</tr>
<tr>
<td>CONTRAST (ALPHA = 3, THETA = 4)</td>
<td>26.0</td>
</tr>
</tbody>
</table>

*Table 2-I: Range of Thresholds of Connectivity*

Local connectivity immediately discredited the distinct model of similarity. Upon comparison to the matches produced by the other models, those of the distinct model seemed almost arbitrary. Descriptions of items that are normally classified into separate categories are all linked together by the distinct model. The common model did fairly well when compared to the ratio and contrast models, but failed in some cases by matching objects solely because of their large descriptions. The ratio and contrast models (specifically the contrast model with theta = 4, alpha = 2, and beta = 1) were nearly indistinguishable. Only on a few matches did they disagree, and these were very similar in any case.
The comparison of the two methods of determining thresholds to ideal thresholds showed that they, too, performed nearly identically. The average size method came as close to the ideal thresholds as did the average depth of connectedness method, and did so with the same average size for each domain (see table 2-II and table 2-III). This is surprising for two reasons. First, the domains are quite different in size. Thus it would seem that the average number of descriptions per piece would also be quite different. Second, the average depth of connectedness within a piece seems to me to be a much more justifiable measure of the quality of that piece than the number of members that it has.

### RATIO MODEL

<table>
<thead>
<tr>
<th>DOMAIN</th>
<th>THRESHOLD (for size = 3)</th>
<th>IDEAL THRESHOLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>FURNITURE</td>
<td>.23</td>
<td>.22</td>
</tr>
<tr>
<td>TRANSPORTATION</td>
<td>.22</td>
<td>.23</td>
</tr>
<tr>
<td>FOOD</td>
<td>.23</td>
<td>.24</td>
</tr>
</tbody>
</table>

**CONTRAST MODEL, ALPHNA = 2, THETA = 4**

<table>
<thead>
<tr>
<th>DOMAIN</th>
<th>THRESHOLD (for size = 3)</th>
<th>IDEAL THRESHOLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>FURNITURE</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>TRANSPORTATION</td>
<td>-8</td>
<td>-6</td>
</tr>
<tr>
<td>FOOD</td>
<td>-6</td>
<td>-6</td>
</tr>
</tbody>
</table>

**Table 2-II: Resulting Thresholds from Average Size**

The connectivity and local connectivity experiments showed that the common and the distinct models are inferior models of similarity. They also showed that the contrast and the ratio models are similar in performance, and that both would be useful in the construction of similarity networks. Because of their close
RATIO MODEL

<table>
<thead>
<tr>
<th>DOMAIN</th>
<th>THRESHOLD</th>
<th>IDEAL THRESHOLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>(for depth = .6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FURNITURE</td>
<td>.23</td>
<td>.22</td>
</tr>
<tr>
<td>TRANSPORTATION</td>
<td>.25</td>
<td>.23</td>
</tr>
<tr>
<td>FOOD</td>
<td>.24</td>
<td>.24</td>
</tr>
</tbody>
</table>

CONTRAST MODEL, ALPHA 2, THETA = 4

<table>
<thead>
<tr>
<th>DOMAIN</th>
<th>THRESHOLD</th>
<th>IDEAL THRESHOLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>(for depth = .6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FURNITURE</td>
<td>-1</td>
<td>-2</td>
</tr>
<tr>
<td>TRANSPORTATION</td>
<td>-8</td>
<td>-6</td>
</tr>
<tr>
<td>FOOD</td>
<td>-5</td>
<td>-6</td>
</tr>
</tbody>
</table>

Table 2-III: Resulting Thresholds from Average Depth performance, and because of the different types of links that they produce - contrast model links are directional and thus unsymmetric, while ratio model links are undirected and symmetric - both models were used in the following experiments. The method used to determine the proper threshold in the following work was the average depth of connectedness of a piece. This choice was made because the average depth of connectedness method seems intuitively more justifiable than the average population method.
Chapter 3

Searching a Similarity Network

3.1 Background

The basic method used for searching a similarity network was the hillclimbing search [Winston 84]. Hillclimbing was chosen because it is simple and because it can use the similarity scores to guide its search. The problems seen in the use of hillclimbing for parameter optimization - such as the foothill or the plateau problems - are not a concern, because there is a definite goal. Two kinds of hillclimbing, one with a memory and one without a memory, were used.

An extension to the basic hillclimbing algorithms was implemented in hopes of improving the search’s efficiency. This extension added the following heuristic: if a description along the search path is similar to the goal description, then proceed as in basic hillclimbing by traveling to the node with the next highest similarity score; if the current description and the goal description are dissimilar, however, go to the description with the lowest similarity score. If the goal is similar to the current description, then traveling to a node with a high similarity score should also produce a close match to the goal by the limited transitivity of similarity. If, on the other hand, the goal is dissimilar to the current description, then it is best to continue the search at a node which is also dissimilar to the current description. I called this heuristic searching by match scores.
3.2 Experiments

Both the contrast and the ratio versions of similarity networks were used for testing the hillclimbing searches. The actual experiment involved running a Monte Carlo simulation for each of the searches on the different networks. Approximately 170 searches were run on each network. This was by no means exhaustive, but as per the Monte Carlo method, enough searches were run to reach some sort of statistical asymptote.

3.3 Results

The results of the Monte Carlo simulations for each search is listed in table 3-I and table 3-II. The hillclimbing search without memory is not included, because it did not usually terminate. The reason for this was that the search was conducted over a network and not a tree - therefore the previous node searched was also listed as a child of the current node, and because hillclimbing uses all of these children in sorted order in its search. Thus, when the hillclimbing search without memory travels along a high scoring link, the chances are that it will return back across the very same link on its next move.

Hillclimbing with memory always terminated, but did not guarantee success in all cases. In particular, when searching through the directional links produced by the contrast model the search was able to come to a dead end within a piece, even though that piece had not been entirely searched. When a search ended in failure, its search path was considered to be empty. This explains why the average length of the search paths for the contrast model networks is shorter than those for the networks created by the ratio model.

The use of match scores as an extension to memory-enriched hillclimbing did
RATIO MODEL

<table>
<thead>
<tr>
<th>DOMAIN</th>
<th>AVE. LENGTH OF SEARCH PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>FURNITURE</td>
<td>9.78</td>
</tr>
<tr>
<td>TRANSPORTATION</td>
<td>15.91</td>
</tr>
<tr>
<td>FOOD</td>
<td>24.38</td>
</tr>
<tr>
<td>OVERALL</td>
<td>16.69</td>
</tr>
</tbody>
</table>

CONTRAST MODEL, ALPHA = 2, THETA = 4

<table>
<thead>
<tr>
<th>DOMAIN</th>
<th>AVE. LENGTH OF SEARCH PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>FURNITURE</td>
<td>8.25</td>
</tr>
<tr>
<td>TRANSPORTATION</td>
<td>13.80</td>
</tr>
<tr>
<td>FOOD</td>
<td>12.49</td>
</tr>
<tr>
<td>OVERALL</td>
<td>11.51</td>
</tr>
</tbody>
</table>

Table 3-I: Results of Searching with a Memory

RATIO MODEL

<table>
<thead>
<tr>
<th>DOMAIN</th>
<th>AVE. LENGTH OF SEARCH PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>FURNITURE</td>
<td>9.35</td>
</tr>
<tr>
<td>TRANSPORTATION</td>
<td>15.88</td>
</tr>
<tr>
<td>FOOD</td>
<td>24.47</td>
</tr>
<tr>
<td>OVERALL</td>
<td>16.57</td>
</tr>
</tbody>
</table>

CONTRAST MODEL, ALPHA = 2, THETA 4

<table>
<thead>
<tr>
<th>DOMAIN</th>
<th>AVE. LENGTH OF SEARCH PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>FURNITURE</td>
<td>8.12</td>
</tr>
<tr>
<td>TRANSPORTATION</td>
<td>14.10</td>
</tr>
<tr>
<td>FOOD</td>
<td>12.52</td>
</tr>
<tr>
<td>OVERALL</td>
<td>11.58</td>
</tr>
</tbody>
</table>

Table 3-II: Results of Searching by Match Scores

...
this is not due to a flaw in the similarity information or in the extension heuristic, but from the fact that the descriptions clustered in one piece (that are children of each other) are all very similar. The use of the high threshold necessary to optimize the pieces also filters out any dissimilar descriptions that would have made this extension effective.
Chapter 4

Representing a Similarity Network

4.1 Background

There are two motivations behind representing similarity networks. First, the representation will include information on the pieces of the network that can be used to increase the efficiency of searching a similarity network. Second, the representation will link multiple similarity networks to allow the searching among these networks. This representation will be made general enough to later allow for multiple layers of similarity networks, or networks of networks.

The representation used for a similarity network consists of three parts, a network flag, a list of neighboring networks, and a list of the pieces in the network. The network flag indicates whether the object represented is a network or just a piece of a network. This flag will allow pieces to be changed to networks easily when layered networks are possible and when large pieces may be made into subnetworks. The list of neighboring networks is used to access one of multiple networks when the search determines that it is finished with the current network. The list of pieces contains the names of the pieces contained in the network. Each piece is represented separately, using a representation similar to that of the parent network.

Representations of pieces consist of four parts. First is a flag indicating that the object is a piece and not a network. Second is the name of the parent network that the piece is a member of. Third is a list of the elements of the piece. These elements are the actual descriptions of the objects contained in the network. Fourth
is the name of the prototypical member of the piece. The prototypical member is found by matching each description in the piece to all of the other descriptions and summing the match scores obtained for that description. The description with the highest total score is designated the prototypical member for that piece. There are other methods of determining a prototype, such as forming a conjunction of all of the features of the descriptions of the piece or choosing the most frequently mentioned features in the piece, that are no doubt better models from a psychological point of view [Reed 82]. The summing of match scores, however, is easy to implement and uses the features of the descriptions for matching. It seems reasonable, therefore, to use the summing scheme to determine a prototypical member. It is this prototypical member that provides the similarity information that allows for the increased efficiency of the searches.

The efficiency increase in the searches results from sorting the pieces according to how well their prototypical members match the goal object. This increases efficiency because the members of each piece are very similar, and the prototypical member is representative of that piece.

Two searches were developed to use the representation of similarity networks. The first used the sorted pieces heuristic to enhance the hillclimbing with memory search described in Chapter 2. The sorted pieces search begins by matching the goal node to each of the pieces' prototypes. The pieces are then sorted according to the match score produced by its prototype and the goal. During the search, when the current piece has been completely examined, the search gets its next piece from the list of sorted pieces. Another search, based upon the search using sorted pieces, was developed to use the list of neighboring networks in the representation to travel among multiple networks. Figure 4-1 shows the relationship that the searches have to each other. Using the list of neighbors, the final search has access to the
similarity networks in each of the previously described domains: furniture, transportation, and food.

\[\text{search among multiple networks}\]
\[\text{search by sorted pieces}\]
\[\text{(search by match scores)}\]
\[\text{hillclimbing with memory}\]
\[\text{hillclimbing without memory}\]

**Figure 4-1:** The Searches Developed

### 4.2 Experiments

The search based upon the sorted pieces heuristic was tested in the same manner as the basic hillclimbing searches described in Chapter 2, via a Monte Carlo simulation. The search designed to traverse multiple networks was tested on just a few examples. This was done to show that multiple network search was possible, not that it was efficient.
4.3 Results

The results of the Monte Carlo simulation on the hillclimbing with sorted pieces are shown in 4-I. As expected, the use of sorted pieces greatly increased the efficiency of the search in terms of the length of the search path. However, the matches necessary to sort the pieces increased the time necessary to conduct the search. No actual measure of the time increase was made, but a definite tradeoff exists between efficiency in terms of the length of the search path and efficiency in terms of the time necessary to conduct the search.

<table>
<thead>
<tr>
<th>RATIO MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOMAIN</td>
</tr>
<tr>
<td>FURNITURE</td>
</tr>
<tr>
<td>TRANSPORTATION</td>
</tr>
<tr>
<td>FOOD</td>
</tr>
<tr>
<td>OVERALL</td>
</tr>
</tbody>
</table>

CONTRAST MODEL, ALPHA = 2, THETA = 4

<table>
<thead>
<tr>
<th>CONTRAST MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOMAIN</td>
</tr>
<tr>
<td>FURNITURE</td>
</tr>
<tr>
<td>TRANSPORTATION</td>
</tr>
<tr>
<td>FOOD</td>
</tr>
<tr>
<td>OVERALL</td>
</tr>
</tbody>
</table>

Table 4-I: Results of Searching with Sorted Pieces

The search among multiple networks was shown to work, but was not tested for efficiency. At present, the search among networks exhaustively searches one network before going on to the next. The efficiency of this search may be improved in two ways. First, instead of just sorting the pieces of a network, pieces scoring
below a threshold may be dropped from the search entirely. Excluding pieces from the search removes any guarantee of success that it might have had, but it would prevent the search from looking through the network exhaustively. Second, networks themselves may be given a prototype derived from the prototypical members of its pieces. In this way, networks may also be sorted by matching the goal description to each network's prototype. The result of this sorting would be similar to that obtained by sorting pieces.
Chapter 5

Future Directions

One potential extension to the work done on similarity networks is a study of multiple layers of similarity networks. This should include the extensions mentioned in the results section of Chapter 3 in order to make the searching of multiple layers more efficient. The study of multiple layers might also include work on forming subnetworks from pieces when that is appropriate. For example, the food domain from my research contains a large number of items. If even more were added, some of the pieces would grow enough (such as the piece containing the vegetables or the one with the dairy products) to require being made into a subnetwork. There should be a way to decide automatically when this is necessary, and then use the representation mechanism to form the new network. Multiple layers of similarity networks would also allow multiple similarity networks to be grouped in a meaningful way.

Another important extension to similarity is the ability to index new descriptions into the networks. This could be accomplished by searching the similarity networks for the best match possible, and placing the piece in that network. A new description would then be matched to the other descriptions in the network to find the new links. This extension would make it possible to use similarity networks as a part of an expanding system.

Similarity networks might also be used for purposes other than simply storing descriptions of objects. They might also be used to find objects that would serve as substitutes for other objects. Substitutes are necessary when the actual object that is desired is unavailable. A substitute could be found by searching the similarity
networks for as close a match as possible. Substitution would be more accurate if features were not all weighted the same by the matcher, and the important features could be searched for.
Chapter 6

Conclusions

An important part of any memory system is the ability to retrieve the items that are stored in the system quickly and efficiently. Similarity networks provide one means of accomplishing this.

Similarity networks can be automatically constructed given the descriptions of the objects to be placed in the network. The contrast and ratio models of similarity are good models to use because they provide a good range of choices of a threshold and because they closely match intuitive ideas on which items are similar. The choice of whether to use the ratio model or the contrast model depends upon the type of links in the network that are desired. For undirected links, the ratio model should be used. For directed links, the contrast model with some focusing (alpha > 1) should be used. The proper threshold at which similarity scores are made into links may be determined through the depth of connectedness of the pieces that the threshold produces. The similarity networks constructed in this manner are good (a subjective measure) and preserve a lot of similarity information that may be used in searching the networks.

The best method of searching within a single network relies on an extension to the hillclimbing search with a memory. This extension involves sorting the pieces of the network to be searched by matching the goal description against the prototypical member of each piece. Sorting pieces greatly reduces the number of nodes searched, but more time than the simple hillclimbing with memory search because of all of the matching that it has to do. An extension involving the use of match scores was not effective, because the members of each piece are very similar.
It is also possible to search among multiple similarity networks by using the representation developed for similarity networks. This multiple network search is not efficient at present, because it exhaustively searches each network before going on to the next. The efficiency of multiple network search could be improved by using techniques similar to the ones used in sorting pieces, however. This search could easily be extended to work for multiple layers of similarity networks as well.

Multiple layers of similarity networks are an extension to the current work on similarity networks that would allow for the making of large pieces to be made into subnetworks. Multiple layers would also allow multiple similarity networks to be grouped in a meaningful way.

Further extensions to similarity networks would make them useful for more than just the storage of descriptions. Similarity networks could be used to find substitutes for objects that are not available.

Similarity networks are an efficient and effective means of storing descriptions. Not only may they be constructed and represented automatically, but they may also be searched easily. Similarity networks are useful for storing descriptions of objects in a meaningful way. Not only may objects be retrieved efficiently, but other operations on them are possible as well.
Appendix A
The Description of an Armchair

(start-story 'armchair)

[armchair about armchair_1]
[armchair about person_1]
[armchair about person_back_1]
[armchair about person_arms_1]

[armchair_1 ako armchair]

[[armchair_1 ako armchair]  
because
[[person_1 sit *] on armchair_1]]

[[armchair_1 ako armchair]  
because
[[person_1 rest person_back_1] on armchair_1]]

[[armchair_1 ako armchair]  
because
[[person_1 rest person_arms_1] on armchair_1]]

(index 'armchair)
Appendix B

The Similarity Network Produced for the Furniture Domain

(show-net furniture)

Similarity threshold is 0.23
Alpha is 1
Beta is 1
Theta is 1
CHAIR links... ((ARMCHAIR 4\13) (ROCKING_CHAIR 1\4)
(SOFA 2\7) (PARK_BENCH 2\7))

ARMCHAIR links... ((CHAIR 4\13) (ROCKING_CHAIR 6\17)
(SOFA 4\17) (PARK_BENCH 4\17))

ROCKING_CHAIR links... ((CHAIR 1\4) (ARMCHAIR 6\17)
(SOFA 1\3) (PARK_BENCH 1\3))

TABLE links... NIL
END_TABLE links... ((COUNTER 4\15))

DESK links... NIL
COUNTER links... ((END_TABLE 4\15))

SOFA links... ((CHAIR 2\7) (ARMCHAIR 4\17)
(ROCKING_CHAIR 1\3) (BED 1\3)
(PARK_BENCH 3\8) (BALL_PARK_BENCH 1\4))

BED links... ((SOFA 1\3) (PARK_BENCH 1\4)
(BALL_PARK_BENCH 2\7))

PARK_BENCH links... ((CHAIR 2\7) (ARMCHAIR 4\17)
(ROCKING_CHAIR 1\3) (SOFA 3\8)
(BED 1\4) (BALL_PARK_BENCH 1\3))

BALL_PARK_BENCH links... ((SOFA 1\4) (BED 2\7)
(PARK_BENCH 1\3))

STOOL links... NIL
BUREAU links... ((CHINA_CABINET 3\11) (NIGHTSTAND 3\11))
CHINA_CABINET links... ((BUREAU 3\11) (CUPBOARD 3\11)
(NIGHTSTAND 3\13))

CUPBOARD links... ((CHINA_CABINET 3\11))
SHELF links... NIL
NIGHTSTAND links... ((BUREAU 3\11) (CHINA_CABINET 3\13))

;; The pieces of the furniture network

(connectivity furniture)

(((CHAIR) (ARMCHAIR 4\13) (ROCKING_CHAIR 1\4) (SOFA 2\7)
(PARK_BENCH 2\7) (BED 1\3) (BALL_PARK_BENCH 1\4))

((TABLE)))
((END_TABLE) (COUNTER 4\15))

((DESK))

((STOOL))

((BUREAU) (CHINA_CABINET 3\11) (NIGHTSTAND 3\11) (CUPBOARD 3\11))

((SHELF)))
Appendix C
A Connectivity Histogram for Furniture

Connectivity histogram for similarity model RATIO.
...Plotting the connectivity for a given threshold.
0    *
0.025 *
0.05  *
0.075 **
0.1   **
0.125 **
0.15  **
0.175 ***
0.2   ****
0.225 ********
0.25  ********
0.275 *********
0.3   *********
0.325 *********
0.35  ***************
0.375 ******************
0.4   ******************
0.425 ******************
0.45  ******************
0.475 ******************
0.5   ******************
Appendix D
Local Connectivity for Furniture

;; Ratio Model, N = 2

CHAIR links... ((ARMCHAIR 4\13) (SOFA 2\7))
ARMCHAIR links... ((ROCKING_CHAIR 6\17) (CHAIR 4\13))
ROCKING_CHAIR links... ((ARMCHAIR 6\17) (SOFA 1\3))
TABLE links... ((END_TABLE 2\9) (DESK 5\23))
END_TABLE links... ((COUNTER 4\15) (TABLE 2\9))
DESK links... ((TABLE 5\23) (END_TABLE 1\5))
COUNTER links... ((END_TABLE 4\15) (TABLE 4\19))
SOFA links... ((PARK_BENCH 3\8) (ROCKING_CHAIR 1\3))
BED links... ((SOFA 1\3) (BALL_PARK_BENCH 2\7))
PARK_BENCH links... ((SOFA 3\8) (ROCKING_CHAIR 1\3))
BALL_PARK_BENCH links... ((PARK_BENCH 1\3) (BED 2\7))
STOOL links... ((CHAIR 2/9) (BED 2/11))
BUREAU links... ((CHINA_CABINET 3\11) (NIGHTSTAND 3\11))
CHINA_CABINET links... ((BUREAU 3\11) (CUPBOARD 3\11))
CUPBOARD links... ((CHINA_CABINET 3\11) (BUREAU 1/5))
SHELF links... ((BUREAU 1\5) (CUPBOARD 1\5))
NIGHTSTAND links... ((BUREAU 3\11) (CHINA_CABINET 3\13))
Appendix E
The Representation of the Furniture Network

(show-rep 'furniture-net)

The representation of network FURNITURE-NET is as follows:
Neighbors: (TRANSPORTATION-NET FOOD-NET)
Members:
Piece piece-1905
  Prototype: SOFA
  Members: ((CHAIR) (ARMCHAIR -5) (SOFA -6)
          (PARK_BENCH -6) (STOOL -5)
          (ROCKING_CHAIR -5) (BED -5)
          (BALL_PARK_BENCH -5))
  Parent: FURNITURE-NET
Piece piece-1906
  Prototype: TABLE
  Members: ((TABLE))
  Parent: FURNITURE-NET
Piece piece-1907
  Prototype: END_TABLE
  Members: ((END_TABLE))
  Parent: FURNITURE-NET
Piece piece-1908
  Prototype: DESK
  Members: ((DESK))
  Parent: FURNITURE-NET
Piece piece-1909
  Prototype: COUNTER
  Members: ((COUNTER))
  Parent: FURNITURE-NET
Piece piece-1910
  Prototype: BUREAU
  Members: ((BUREAU) (CHINA_CABINET -5)
           (CUPBOARD -6) (SHELF -6)
           (NIGHTSTAND -5))
  Parent: FURNITURE-NET
References

[Brady 84] Michael Brady.  
The Mechanic's Mate.  
ECAI 84, 1984.

Categorization of Natural Objects.  

[Reed 82] Stephen K. Reed.  

[Rosch 78] Eleanor Rosch.  
Principles of Categorization.  

Categories and Concepts.  

[Tversky 77] Amos Tversky.  
Features of Similarity.  

The Psychology of Computer Vision.  

Learning New Principles from Precedents and Exercises: the Details.  
AIM 632, 1981.

Learning Physical Descriptions from Functional Definitions, Examples, and Precedents.  
AIM 679, 1983.

[Winston 84] Patrick H. Winston.  
Artificial Intelligence.  