An Information Retrieval Model for Coordination Systems Based on Fuzzy Proximity Networks*

Chuen-Tsai Sun

Computer Science Division
University of California, Berkeley, CA 94720

Abstract

This article describes the design of a computational model for information retrieval which provides coordination services for collaborative agents. New requirements resulting from recent technological development are identified. Coordination problems in terms of information retrieval are defined. Limitations of previous models based on fuzzy sets are discussed to justify the need for a new approach. We applied results of psychological experiments when making design decisions so that the proposed system simulates human behavior. The design framework, fuzzy proximity networks together with typicality measurement algorithms, is described in detail. We also show how the system takes users' personal profiles into consideration when processing their queries. The weighted query method in traditional non-fuzzy information retrieval systems is just a special case in the proposed model. Learning algorithms are suggested to update the networks and to construct a keyword hierarchy automatically. Finally, a research plan is described.

Introduction

In a distributed processing environment, coordination problems have been receiving more and more attention. To help cooperative agents achieving a common goal with the minimum total cost, a coordinator must be able to evaluate and combine individual requests from the agents. Researchers have been analyzing coordination problems from various angles, e.g., resource allocation, protocol design, and communicational complexity. This project addresses the coordination problem in terms of information retrieval.

Meanwhile, modern memory technology results in massive amount of information accumulated in databases without immediate use. The term *information mining* has been created to pinpoint the major characteristic of database management in this decade. To deal with huge amount of information, retrieval methods with exact keyword matching are not qualified. Results of research on uncertainty can be used to help alleviating this problem. An information mining system must be able to find all *relevant* documents and rank them by their *appropriateness* to the current application. Researchers have been trying various approaches, such as probability-based models, rule-based models, and models based on fuzzy set theory and fuzzy logic, to address the problem. This project focuses on fuzzy models because we think a computational model that pinpoints linguistic fuzziness plays an inevitable role in intelligent, efficient information retrieval.

*Research supported in part by NASA Grant NCC-2-275; LLNL Grant No. ISCR 89-12; MICRO State Program Award No. 89-046; MICRO Industry: Rockwell Grant No. B02302532.

The drawbacks of previous fuzzy extensional models suggest a more representational approach in dealing with the issues. Based on psychological research on human memory structures, we propose a computational model which uses *fuzzy proximity networks* as its knowledge representation. Greedy algorithms are adopted to guarantee efficiency. Inexact keyword matching and multiple queries are handled properly in this model.

Each agent has his own preference in setting the semantic background on which his queries are processed. This desired feature has long been ignored. Research on semantic and episodic memory gives us insight to solve this problem. Information of personal preference is realized by subnetworks called *episodic maps* in our model. Algorithms are provided to trigger episodic maps and to plug them into the entire semantic background.

The semantic background and the episodic maps are subject to changes. A learning algorithm is proposed to train the networks by examples, after the initial values are set by a statistical method. Another algorithm is suggested to automatically construct a concept hierarchy for advanced uses, such as hypertexting.

Coordination Problems: An Example

We use a real case to illustrate problems of coordination in the framework of information retrieval. Two research groups, *housing* and *commuting,* in a city planning institute want to conduct a joint study on the local housing/transportation problems from an aggregate point of view. They need information services in several aspects: (1) document survey, (2) questionnaire design, (3) sample household selection, and (4) data analysis. In other words, at different stages of the joint project, they want to retrieve previous research reports, questionnaire packages, household records, and data sets from databases. These information service duties can be divided into two categories: those for a project coordinator, and those for individual groups. In our example, (2) and (3) are for the coordinator because a single questionnaire and a single mailing list are needed. On the other hand, (1) and (4) are for individual groups because each group needs to study documents and data on their own.

The problem is how to discover all the *relevant* pieces of information for each group, so that the combination of their local decision optimizes the global benefit. The collected information should not only reflect the two groups' common interests, but also build possible connections between them.

For example, each group's interest profile is represented as a query. The emphases of the housing group include *household size*, *housing status, moving record,* etc. On the other hand, the commuting group focuses on *commuting mode, travel time, income*, etc. Note that there may be some shared items, such as *income* and *household size*, which show the common interests of the two groups explicitly. Obviously, data sets including the shared items should be retrieved for their use. Most previous

90CH2930-6/90/0000-0418$01.00 © 1990 IEEE

information retrieval models handle this sort of problem.

However, this is not the whole story. In real world, implicit connections between organizations, persons, or concepts play an important role in achieving the globally optimal solution. For example, *land using* is a strong link between *housing* and *commuting* but neither group is likely to specify items in *land using* such as *housing stock, zoning,* or *development policy*, because these items are not directly related to either group's local problems. However, without *land using* as the common base, it is hard for the two groups to achieve any consistent conclusion. Consequently, it is the duty of an intelligent information system to build bridges between the two collaborative groups. For instance, a data set concerning items in *land using*, should also be adopted for the joint project to get a complete picture of the planning issues. Previous models are weak in this respect.

Previous Models

A general information retrieval system consists of a set D of *documents* and a set K of *keywords*. Documents are indexed with keywords by authors or librarians. When users want to retrieve documents, they use keywords in K to form a query. Ideally, an intelligent system retrieves all *relevant* documents and ranks them according to their *degrees of relevance* to the query. Consequently, users can find out the most interesting materials for their current projects.

If we consider the same problem in a coordination environment, the system should be more powerful since it must be able to manipulate collaborative queries from different users at the same time. Take the joint city planning project as an example. One of the goal is to find the implicit links between the two project groups. Since the interest profile of each group is merely represented with keywords from K, it is the duty of the retrieval mechanism to find the implicit connections.

In the past, fuzzy information retrieval models [14, 16, 3, 9] concentrated on single-user applications. Most models applied *fuzzy set theory* [17]. Briefly speaking, each keyword is considered as a *fuzzy set* to which each document has a *membership* value, between 0 and 1. In other words, the value indicates the degree of belonging of the document to the fuzzy set denoted by the keyword.

A document with multiple keywords is taken as a member of the aggregate fuzzy set defined by the keywords. The document's membership to this aggregate set is calculated by a set of operators. A query is considered as a particular aggregate set in which the *ideal document* in the querier's mind has full membership. Various retrieval mechanisms are used to measure the degree of *similarity* of each document to the ideal one. This is done by algorithms based on fuzzy set combinations. The desired documents are then retrieved according to their similarity measures. Membership values associated with document keywords and query keywords are usually assumed to be assigned subjectively by indexers and users, respectively.

Moreover, to take inexact matching into consideration, researchers extended the basic model by incorporating a *fuzzy thesaurus* [2, 12]. A fuzzy thesaurus is a network-like structure in which nodes denoting related keywords are linked together with (weighted or unweighted) edges. A search process is performed for each query to find all qualified documents. If a keyword K_i is in the query set, then documents with K_i's *broader* or *related* terms should receive partial credit from the retrieval mechanism. Generally speaking, the existing search algorithms are all very time-consuming although the idea of thesaurus is worth studying.

Before discussing the drawbacks of the previous models, we must indicate that it is not natural at all for either an indexer or a user to specify numbers with keywords. Even we accept this unrealistic operational assumption, there are many problems with the models. The first is called *personal difference*, which results from subjectivity of assigning membership values. The same value may have multiple interpretations in different persons' mind. This simple fact limits the models for personal use, such as a bibliographical aid. It is not suitable for multi-user applications, not to mention situations with collaborative agents.

The second issue is named *abstraction-typicality gap*. Researchers observed that a number associated with a query keyword stands for a *typicality* measure, i.e., how typical a document is to the keyword. On the other hand, the same number associated with a document keyword stands for an *abstraction* measure, i.e., the degree of appropriateness for using the keyword in abstracting the document. Thus, any mechanism that treats the same number in both a document and a query as a perfect match is questionable.

The third problem comes when the system tries to combine partial credits of matched keywords into a total. The combination process depends on our understanding of the connectives that people use when they specify a list of keywords. It could be additive, compensatory, or logic [18]. Each interpretation leads to different mechanisms. Generally speaking, it is neither easy to identify the connective people actually use nor proper to explicitly assign one for them to use. we call this phenomenon *semantic ambiguity*.

Moreover, because of their weakness in representation, the previous models could not support coordination services competently. For the first objective of coordination, i.e., to recall information of common interests, what these models can do is to use set intersection to form a narrower query. This is usually not what users exactly want. For the second objective, to discover implicit connections among agents, these models are even less effective. The use of thesauri can help a little here but it involves a tedious search process. A new information processing model should answer the above questions.

Results of Psychological Experiments

It is believed that cybernetical actualization of psychological results could bring considerable effects on intelligent system design. Psychologists have been studying fuzziness for more than two decades [1, 15]. Results of psychological experiments on memory structures, memory recall, similarity, and typicality are directly related to the mental process of information retrieval.

Psychological models to catch fuzziness can be divided into categories according to various criteria. One dimension is *extensional* models versus *representational* models [4]. In extensional models, basic components are treated as sets. The formation of complex concepts is based on operations that combine extensions, e.g., set union and intersection. In representational models, on the other hand, concepts are modeled as structured descriptions. Complex concepts are perceived as aggregate structures.

Experiments showed that extensional models did not catch enough details, in terms of representational granularity, to form integrated fuzzy concepts [6, 8]. Consequently, researchers constructed more representational models, such as various networks, to reflect fuzzy phenomena. Note that information retrieval models introduced in the previous section belong to extensional models. Thus, we are suggested to design a more representational model to replace the old ones.

Other experiments demonstrated that information indexing and retrieval should take personal difference into consideration. In other words, an information server should set customized semantic backgrounds for different users before processing their requests. Psychological studies on *semantic* and *episodic* memories [7], see Figure 1, give us the insight to design an information retrieval model which handles individual emphases better than traditional approaches. In this model, the role of input data is two-fold. It first triggers related episodic memory for the current situation. After the combination of the recalled episode and the general semantic memory, the input is processed in the newly formed working memory.

In summary, based on the results of psychological studies, the following properties are desirable for an intelligent information retrieval system: (1) a representational model, and (2) incorporating episodic information, such as personal difference/preference.

Design Scheme

Here, to answer the above questions, we propose a new information retrieval model based on *fuzzy proximity networks*. First of all, we believe that indexing/querying with keywords but without numbers is appropriate for both authors and queriers. However, instead of focusing on keywords themselves as in previous models, our design scheme emphasizes on keyword structures and connections.

A fuzzy proximity network is a weighted undirected graph, denoted by $G_{FPN} = (N, A)$. Each node (vertex) $i \in N$ represents a keyword. The weight $w(i,j)$ on an arch/edge $(i,j) \in A$ denotes the relevance, a fuzzy relation, between the two nodes i and j. We can define a corresponding *fuzzy distance network* G_{FDN} by replacing $w(i,j)$ by $1 - w(i,j)$.

There are many ways to interpret *relevance* between keywords. We select Miyamoto's *co-occurrence* measure as our operational definition [11]. In brief, the more often two keywords co-occur (appear simultaneously) in a document, the stronger is the connection between them. The co-occurrence algorithm has extended this idea from documents to their backward and forward citations, see Figure 2.

Keywords K_a and K_b occur coincidently in document 1, so they are considered relevant to each other. Further, documents 2 and 3 are in the backward citation (reference) of document 1; document 4 and 5 are in the forward citation (citation index) of document 1. Consequently, keyword pairs K_c–K_d and K_e–K_f are also instances of co-occurrence in a weaker sense. Notice that this extension deals with synonymous terms quite well.

Also notice that this algorithm is just for the initial setting of the weights in our model. Later on, the network can adjust itself through learning algorithms discussed in a separate section.

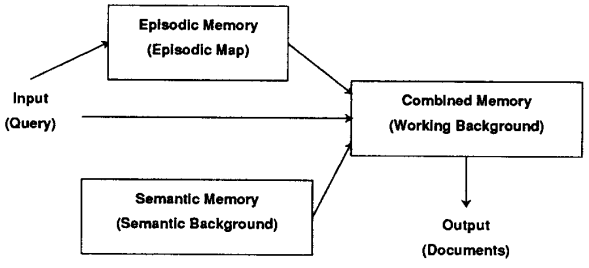

Figure 1: *Episodic and semantic memory.* Terms in parentheses are corresponding concepts in the information retrieval model.

Using a statistical algorithm to set initial values avoids learning from scratch.

Typicality measurement is based on the calculation of the weight of a *maximum spanning tree* in this model. A spanning tree is a tree that covers a given vertex set. The weight of a spanning tree is the sum of the weights on edges in that tree. A maximum spanning tree is a spanning tree with the unique maximum value. We use $MST(V)$ to denote the maximum spanning tree defined by a vertex set V and $W_{MST}(V)$ to denote its weight. Note that conceptually G_{FPN} is a complete graph. Unrelated nodes are assumed to have an edge with weight 0.0 in between. Thus, it is always possible to find a maximum spanning tree given a vertex set.

When doing a query, we use $W_{MST}(Q)$ as the *typicality base*, i.e., the strength of inner connection of the query (or the ideal document represented by the query). The typicality measure $T_Q(D_i)$ of a document D_i to a query Q is calculated by the following formula.

$$T_Q(D_i) = \frac{W_{MST}(Q \cap D_i)}{W_{MST}(Q)} \quad (1)$$

Given a query, we measure the typicality value of each document in a database and then sort the documents into descending order.

The rationale behind this retrieval/ranking mechanism is three-fold. The first reason is that we believe people emphasize the connection between keywords when they specify a query set. In other words, when people name a list of keywords in a query, usually their search target is something that can link these keywords together to form one or more structured concepts. A maximum spanning tree is a natural representation of the strongest connection of the focussed keywords, and its weight is a proper gestalt measure of the strength of the inner connection. The ideal document has typicality measure 1.0, as expected.

The same reason explains why we use $W_{MST}(Q \cap D_i)$ as the numerator in equation 1. We believe that the keyword set $Q \cap D_i$ forms a keyword/concept cluster, so each keyword should not be treated individually as in previous fuzzy or non-fuzzy information retrieval models. On the contrary, the stronger the inner connection in this keyword/concept cluster, the higher its potential to be used as a meaningful component in the querier's project.

When we consider information integration, the use of weight of maximum spanning trees is further justified. For example, in a coordination environment we need to combine queries from collaborative agents to form a joint query set; to put several inter-related documents into an archive file we need to create a joint index set. We believe this sort of combination is not additive. In other words, any mechanism based on keyword set union does not face the fact that the combination of two concepts is a new concept with its own structure and gestalt properties of that structure. We think that the desirable gestalt property of keyword set combination is a possibility measure; in other words, it is subadditive, see [5].

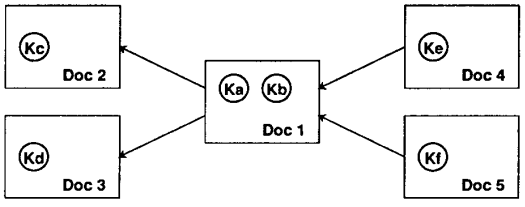

Figure 2: *Co-occurrence of keywords.*

W_{MST} is a subadditive set function defined on the vertex set to be covered. It is easy to be proved given the fact that matroid rank function is subadditive and that proximity network can be considered as a graphic matroid. Thus, when keywords are combined to form a complex concept, we have a measure to reflect the strength of the inner connection of this newly created concept. We assert that this is a much more accurate measure than before, especially when we take information combination into consideration. This is the second reason to use the weight of a maximum spanning tree.

The third consideration is about computational complexity. Since we are trying to manipulate very large amount of documents, the calculation for basic measurement must be efficient. The choice of maximum spanning tree algorithms meets this requirement well. A well-known *greedy algorithm* exists for finding a maximum spanning tree [10]. In summary, this mechanism catches the desired features: representational modelling, subadditive measurement, and computational efficiency.

Two additional points should be made here. First, the measure of inner connection strength depends on the value setting on the proximity network, and the setting may differ from person to person. That is the reason why we must take personal difference/preference into consideration, see the section of *Episodic Maps*.

Second, singleton queries should be treated as an exception in our model. Singleton queries are queries with only one keyword. They emphasizes on the generic meaning of the keyword itself other than its connection to something else. In this case, documents providing either a good introduction or a complete survey to the keyword are the target. Since the interpretation is different, we handle singleton queries separatedly in the proposed system. Two approaches are offered to answer singleton queries, one in the section of *Episodic Maps*, the other one in the section of *Learning Algorithms*.

Inexact Matching

The problem of inexact matching can be easily solved by extending the above mechanism to find *Steiner nodes*. This algorithm is based on the fuzzy distance networks G_{FDN}. A *Steiner tree* covering a vertex set V in G_{FDN} is a tree that covers vertex set $V \cup S, (S \subset N, S \cap V = \emptyset)$, and has the *minimum* weight. Each member of set S is called a Steiner node. The *physical meaning* of a Steiner node is a bypass point that provides a strong connection between two vertices in V.

During retrieval, the calculation can be performed on the subgraph of G_{FPN} defined by $Q \cup D_i$. Hence, for inexact matching, we replace the edge weight $w(i,j)$ in the subgraph by $1 - w(i,j)$ and then find the Steiner tree to cover the vertex set Q. We use S $(S \subset D_i - Q)$ to denote the set of Steiner nodes. The next step is to find a neighboring set S_n of S defined by:

$$S_n = \{n | (m,n) \in A, m \in S, n \in Q - D_i\}. \quad (2)$$

S_n is the counterpart of S in $Q - D_i$, see Figure 3.

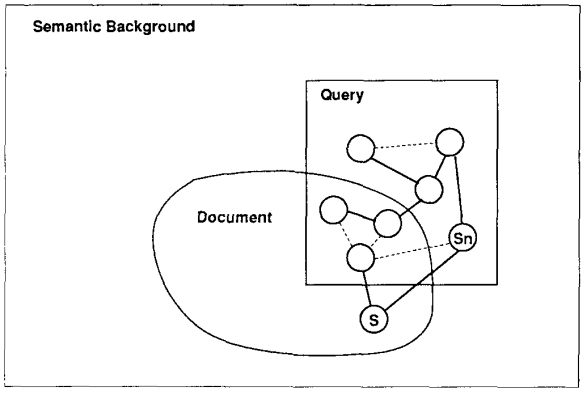

Figure 3: *Steiner nodes in inexact matching.*

In other words, each node in S_n represents a query keywords which has no exact matching in the current document but has a highly relevant counterpart. Obviously, this kind of nodes should be taken into consideration. In this case, typicality of D_i is calculated by:

$$T_Q(D_i) = \frac{W_{MST}((Q \cap D_i) \cup S_n)}{W_{MST}(Q)} \quad (3)$$

Although finding Steiner trees is a NP-hard problem in terms of computational complexity, it does not cause serious trouble in our application. First of all, because of the relatively small cardinality of vertex sets defined by documents and queries, worst-case-exponential algorithms [10] can satisfy our demand. Moreover, there exist polynomial algorithms to find approximate solutions [13]. Basically, they find more Steiner points than the optimal algorithm. However, since the point here is just to find *implicit connections* between a query and a document, more Steiner points means more (but maybe weaker) connections; therefore, these approximate solutions are still very useful for us.

A similar approach is applied to handle queries from collaborative agents. In our housing/commuting joint project example, assume the two project groups specify their focus of study as two queries: Q_h and Q_c. We find a Steiner tree to cover Q_h and Q_c and denote the Steiner set as S. In this case, S represents the implicit connections between the two project groups. Thus, we use $Q_h \cup S$ as the query for the housing group and $Q_c \cup S$ for the commuting group, in applications such as document survey and data analysis. Further, $Q = Q_h \cup Q_c \cup S$ is adopted to be the query for the coordinator to prepare a joint questionnaire and a sample household mailing list.

To this point, we have discussed the basic retrieval mechanism of the proposed model. It is a representational model. The typicality base measure is a subadditive set function. Inexact matching is incorporated by using Steiner trees. The calculation involved is very efficient because it depends on greedy algorithms. Next, we will add personal profiles into this picture.

Episodic Maps

As mentioned before, the strength between two keywords may differentiate from person to person. Ideally, a retrieval system should customize the information background before processing a particular user's query. In our model, each agent's preference, or his *personal profile*, is realized by *episodic maps*.

An *episodic map* is a subgraph of G_{FPN}, the fuzzy proximity network, which is used for an agent's special setting of edge weights, see Figure 4. Each episodic map can be generated with the co-occurrence algorithm by running it on a special set of documents. For example, a city planning researcher may select a bunch of typical articles in his field and run the co-occurrence program to create his own map. The map can also be created subjectively, i.e., a user can set subjective weights on edges of the subgraph defined by the keywords he is interested in.

Relative to episodic maps, we call the entire proximity network the *semantic background* because what it stands for is world-knowledge-like semantics for general public's use. The weights set in the semantic background are considered as de-

fault values. For specific users we want to plug in their personal profiles before processing their queries. When an episodic map is applied the edge weights specified in this map overwrite the ones in the semantic background.

Episodic maps can be called explicitly, i.e., users can customize their query background by specifying whatever episodic maps they want. In this case, episodic maps can be treated as documents stored with a special set of indices. A querier can set his background by specifying some of these special indices to retrieve the suitable maps.

On the other hand, a more interesting approach is to trigger episodic maps by using a query directly. This procedure is analogical to human behavior that is exposed in previous psychological studies. Please refer to Figure 1 again. The triggered maps are those in which the query has high typicality measures. In other words, the following formula is used to evaluate the appropriateness of an episodic map E_i with regard to a query Q.

$$T_{E_i}(Q) = \frac{W_{MST}(Q \cap E_i)}{W_{MST}(E_i)} \tag{4}$$

We then plug the episodic maps into the semantic background and use the incorporated network as our working memory.

We can set a threshold value in selecting episodic maps. If no map has a typicality measure larger than the threshold, we just use the default semantic background. When more than one episodic maps are qualified, they can be plugged into the semantic background sequentially. For example, if we ask the system to pick up the top three episodic maps, E_1, E_2 and E_3, which are all beyond the threshold value, we plug E_3 first into the semantic background, then E_2, then E_1.

When collaborative agents are involved in the querying process, multiple episodic maps are triggered parallelly. In this case these maps are combined into one. If they are overlapping, the *maximum* operator is used on the overlapping part because the system emphasizes on their common interests.

Representational power is not the only benefit of episodic maps. They have also the advantage of easy construction/learning because of their relatively small size compared to the entire network. Moreover, episodic maps can be structured into a hierarchy which is important when processing categoric information, see the next section.

Now we show that the weighted query method in traditional non-fuzzy information retrieval systems is just a special case in our model. Given a weighted query list:

$$(K_1, w_1), (K_2, w_2), \ldots, (K_n, w_n), \tag{5}$$

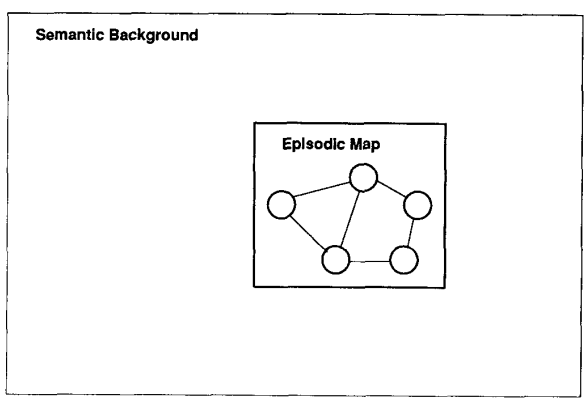

Figure 4: *An episodic map on the semantic background.*

where K_i is a keyword and w_i is the associated weight, we first construct an episodic map $E_{wq} = (N_{wq}, A_{wq})$ with

$$N_{wq} = \{K_1, K_2, \ldots, K_n, D\}, \tag{6}$$

where D is a dummy node, and

$$A_{wq} = \{(K_i, D) \mid K_i \in N_{wq}, K_i \neq D, w(K_i, D) = w_i\}. \tag{7}$$

We add D to each document and plug E_{wq} into the semantic background. Then, we use N_{wq} as our query set. It is easy to show that our retrieval/ranking algorithm calculates exactly the same score for each document as in traditional non-fuzzy models. Consequently, a traditional weighted query is just a special case in our model and can be treated as a special episodic map.

Further, singleton queries can be handled here by treating it as a weighted query with weight 1.0. Another, more interesting, method is shown in the next section.

Learning Algorithms

It is important for the networks to learn so that they reflect the most updated information. Since the semantic background can be considered as a special case of an episodic map, we will discuss the learning algorithms for episodic maps only.

In terms of learning, there are two things the system can do to improve itself. The first one is to adjust the weights on edges so that a map provides a more accurate typicality base for measurement. The second one is to derive an episodic hierarchy out of the maps.

A learn-by-example algorithm is proposed here to adjust the weights on the edges in an episodic map E. A set of exemplar documents are selected. Human experts give each document a numeric value, between 0 and 1, for the typicality measure of the document to this episodic map E. We use $T_E^d(D_i)$ to denote the desired typicality measure of document D_i.

While $T_E^d(D_i)$ is greater than $T_E(D_i)$, i.e., the typicality measure calculated by the system, we can enlarge the calculated measure by increasing weights in $MST(E \cap D_i)$ and/or by decreasing weights in $MST(E)$. We use the following notations in the learning formula.

$$C_a = \mid E \cap D_i \mid -1 \tag{8}$$

$$C_b = \mid E \mid -1 \tag{9}$$

$$MST(E \cap D_i) = \{a_1, a_2, \ldots, a_{C_a}\}, w_{a_1} \leq w_{a_2} \leq \ldots \leq w_{a_{C_a}} \tag{10}$$

$$MST(E) = \{b_1, b_2, \ldots, b_{C_b}\}, w_{b_1} \geq w_{b_2} \geq \ldots \geq w_{b_{C_b}} \tag{11}$$

$$W_a = W_{MST}(E \cap D_i) \tag{12}$$

$$W_b = W_{MST}(E) \tag{13}$$

$$\Delta R = T_E^d(D_i) - T_E(D_i) \tag{14}$$

Now, if $C_a \geq \Delta R \times W_b \times 10$, we adjust the weight w_{a_i} of each edge $a_i \in \{a_1, a_2, \ldots, a_m\}$, where $m = Int(\frac{\Delta R \times W_b \times 10}{C_a})$, with the following formula:

$$w_{a_i}^{t+1} = min(1, w_{a_i}^t + 0.1). \tag{15}$$

If $C_a < \Delta R \times W_b \times 10$, we apply equation 15 to each $a_i \in MST(E \cap D_i)$. Besides, we adjust each edge $b_i \in \{b_1, b_2, \ldots, b_n\}$, where $n = Int(\frac{W_b \times (\Delta R \times W_b \times 10 - C_a)}{C_a})$, with the following formula:

$$w_{b_i}^{t+1} = max(0, w_{b_i}^t - 0.1). \tag{16}$$

This is a monotonic increasing adjustment. It is easy to prove the convergence of the formula. We run this procedure iteratively until the episodic map is tuned to the desired value.

When $T_E^d(D_i)$ is less than $T_E(D_i)$, an adjustment symmetric to the above procedure is applied. Other effective and efficient learning algorithms are still under development. Their performance will be compared so that the best one will be selected for the final implementation.

The second self-construction mechanism, as mentioned before, regards episodic hierarchy. This mechanism is first motivated by the natural need to use episodes as keywords. In other words, episodes are concepts, too. For example, the term *city planning* can be treated as an episodic map by a city planning professional, or as a single keyword by someone who is interested in connecting *city planning* to other fields. Second, concept structures become more and more important for modern information applications, such as hypertexting.

Assume one episodic map E_j is included in another one E_i. A reduced map E_i^* can be created to have E_j represented as a single node, see Figure 5. We offer an algorithm to calculate the weights between this surrogate node E_j and each of its neighboring nodes in E_i^*.

Let n be a node belonging to $E_i - E_j$ and adjacent to at least one node in E_j. We use H to denote the set of neighboring nodes of n in episodic map E_j. The following formula calculates the desired edge weight of $w(n, E_j)$.

$$w(n, E_j) = \frac{W_{MST}(n \cup H)}{W_{MST}(n \cup E_j)} \tag{17}$$

Thus, if a query includes E_j as a keyword, but not the details in the corresponding episodic map, the reduced map E_i^* can be triggered with a higher score than both E_i and E_j so as to provide a more accurate profile. Another use of reduced maps regards to singleton queries mentioned before. If the only query term E is an episode, e.g., *city planning*, we can use the corresponding episodic map as our query. In this case, we expand the simple query to include details. If the keyword is not an episode, we treat the query as a weighted query, as mentioned before.

Concluding Remarks

We proposed a computational model for information retrieval based on psychological and computational considerations. Consequently, it has deeper representative capability and better computational efficiency than previous extensional models. In brief, whereas previous models focus on keywords themselves, our model emphasizes on the connections between keywords. This design choice makes our model strong in building bridges among collaborative agents. Besides, it possesses good features such as subadditivity, personal profiles, and learning capability.

The model has been tested on several small cases, such as the joint city planning project mentioned before. It showed theoretic soundness as well as retrieval power in finding implicit connections. The next step is to build a full-scale system.

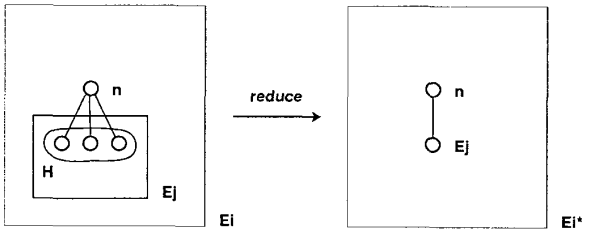

Figure 5: *Reduced episodic map.*

A natural expansion of this model is to include applications that take advantage of the episodic hierarchy built by the reduction algorithm. The episodic hierarchy provides a solid base for dynamic linkage of information. By intuition, modern computer applications, such as hypertexting, can be benefited from the proposed model. Further, this model has the potential to be the information server for distributed artificial intelligence (DAI) systems.

References

[1] A.N. Averkin and V.B. Tarasov. The fuzzy modelling relation and its application in psychology and artificial intelligence. *Fuzzy Sets and Systems*, 22:3–24, 1987.

[2] James C Bezdek, Gautam Biswas, and Li-Ya Huang. Transitive closures of fuzzy thesauri for information-retrieval systems. *Int. J. Man-Machine Studies*, 25:343–356, 1986.

[3] Duncan A. Buell. An analysis of some fuzzy subset applications to information retrieval systems. *Fuzzy Sets and Systems*, 7:35–42, 1982.

[4] Benjamin Cohen and Gregory L. Murphy. Models of concepts. *Cognitive Science*, 8:27–58, 1984.

[5] Didier Dubois and Henri Prade. *Possibility Theory: An Approach to Computerized Processing of Uncertainty.* Plenum Press, 1986.

[6] Harry M. Hersh and Alfonso Caramazza. A fuzzy set approach to modifiers and vagueness in natural language. *Journal of Experimental Psychology: General*, 105(3):254–276, 1976.

[7] Michael S. Humphreys, John D. Bain, and Ray Pike. Different ways to cue a coherent memory system: a theory for episodic, semantic, and procedural tasks. *Psychological Review*, 96(2):208–233, 1989.

[8] Paul Kay and Chad K. McDaniel. The linguistic significance of the meanings of basic color terms. *Language*, 54(3):610–646, 1978.

[9] Etienne E. Kerre, Rembrand B. R. C. Zenner, and Ritaa M. M. De Caluwe. The use of fuzzy set theory in information retrieval and databases: a survey. *Journal of the American Society for Information Science*, 37(5):341–345, 1986.

[10] Eugene L. Lawler. *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart and Winston, 1976.

[11] Sadaaki Miyamoto, Teruhisa Miyake, and Kazuhiko Nakayama. Generation of a pseudothesaurus for information retrieval based on coocurrences and fuzzy set operations. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-13(1):62–70, 1983.

[12] Sadaaki Miyamoto and K. Nakayama. Fuzzy information retrieval based on a fuzzy pseudothesaurus. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-16(2):278–282, 1986.

[13] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry: an Introduction.* New York: Springer-Verlag, 1985.

[14] Tadeusz Radecki. Mathematical model of information retrieval system based on the concept of fuzzy thesaurus. *Information Processing and Management*, 12(5):313–318, 1976.

[15] Michael Smithson. Fuzzy set theory and the social sciences: the scope for applications. *Fuzzy Sets and Systems*, 26:1–21, 1988.

[16] Valiollah Tahani. A conceptual framework for fuzzy query processing – a step toward very intelligent database systems. *Information Processing and Management*, 13:289–303, 1977.

[17] Lotfi. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[18] H. J. Zimmermann and P. Zysno. Decisions and evaluations by hierarchical aggregation of information. *Fuzzy Sets and Systems*, 10:243–260, 1983.