Inference Networks for Document Retrieval

Howard Turtle and W. Bruce Croft
Computer and Information Science Department
University of Massachusetts

COINS Technical Report 90-07
February 1990

Inference Networks for Document Retrieval

Howard Turtle* and W. Bruce Croft
Computer and Information Science Department
University of Massachussetts
Amherst, MA 01003

Abstract

The use of inference networks to support document retrieval is introduced. A network-based retrieval model is described and compared to conventional probabilistic and Boolean models.

1 Introduction

Network representations have been used in information retrieval since at least the early 1960's. Networks have been used to support diverse retrieval functions, including browsing [36], document clustering [6], spreading activation search[4], support for multiple search strategies [8], and representation of user knowledge [26] or document content [37].

Recent work has suggested that significant improvements in retrieval performance will require techniques that, in some sense, "understand" the content of documents and queries [39,7] and can be used to infer probable relationships between documents and queries. In this view, information retrieval is an inference or evidential reasoning process in which we estimate the probability that a user's information need, expressed as one or more queries, is met given a document as "evidence." Network representations show promise as a mechanism for inferring these kinds of relationships [9,4].

The idea that retrieval is an inference or evidential reasoning process is not new. Cooper's logical relevance [5] is based on deductive relationships between representations of documents and information needs. Wilson's situational relevance [40] extends this notion to incorporate inductive or uncertain inference based on the degree to which documents support information needs. The techniques required to support these kinds of inference are not unlike those used in expert systems that reason under uncertainty. A number of competing inference models have been developed for these kinds of expert systems [17,21] and several of these models can be adapted to the document retrieval task.

In the research described here we adapt an inference network model to the retrieval task. The use of the model is intended to:

- Support the use of multiple document representation schemes. Research has shown that a given query will retrieve different documents when applied to different document representations [24,18]. We expect that combining document sets retrieved from different representations will improve recall and that combining the retrieval scores from the different representations will improve precision.

- Allow results from different queries and query types to be combined. Different retrieval strategies will retrieve different documents for the same query, even when the overall performance of the different strategies is the same [7]. We expect that retrieval performance will improve if we combine the results from different strategies and from different queries that express the same information need.

- Facilitate flexible matching between the terms or concepts mentioned in queries and those assigned to documents. The poor match between the vocabulary used to express queries and the vocabulary used to represent documents appears to be a major cause of poor recall [16]. We expect that recall can be improved by allowing more flexible match between query and repre-

*On leave from OCLC Online Computer Library Center

sentation concepts without significantly degrading precision.

In what follows we briefly review candidate inference models, present a formal retrieval model that uses multiple document and query representations to estimate the probability that a document satisfies a user's information need, and compare this model to current retrieval models.

2 Inference networks

The development of automated inference techniques that accomodate uncertainty has been an area of active research in the artificial intelligence community, particularly in the context of expert systems [17,21]. Popular approaches include those based on purely symbolic reasoning [3,12,11], fuzzy sets [41,42], and a variety of probability models [25,2]. Two inference models based on probabilistic methods are of particular interest: Bayesian inference networks [27,20] and the Dempster-Shafer theory of evidence [10,31,32,43].

A Bayesian inference network is a directed, acyclic dependency graph in which nodes represent propositional variables or constants and edges represent dependence relations between propositions. If a proposition represented by a node p "causes" or implies the proposition represented by node q, we draw a directed edge from p to q. The node q contains a matrix that specifies $P(q|p)$ for all possible values of the two variables. When a node has multiple parents, the matrix specifies the dependence of that node on the set of parents (π_q) and characterizes the dependence relationship between that node and all nodes representing its potential causes. Given a set of prior probabilities for the roots of the DAG, these networks can be used to compute the probability or degree of belief associated with all remaining nodes.

Different restrictions on the topology of the network and assumptions about the way in which the connected nodes interact lead to different schemes for combining probabilities. In general, these schemes have two components which operate independently: a *predictive* component in which parent nodes provide support for their children (the degree to which we believe a proposition depends on the degree to which we believe the propositions that might cause it), and a *diagnostic* component in which children provide support for their parents (if our belief in a proposition increases or decreases, so does our belief in its potential causes). The propagation of probabilities through the net can be done using information passed between adjacent nodes.

While not originally cast as a network model, the Dempster-Shafer theory of evidence can be used as an alternative method for evaluating these kinds of probabilistic inference networks. Rather than computing the belief associated with a query given a set of evidence, we can view Dempster-Shafer as computing the probability that the evidence would allow us to prove the query. The degree of support parameters associated with the arcs joining nodes are not interpreted as conditional probabilities, but as assertions that the parent node provides support for the child (is *active*) for some proportion p of the time and does not support the child for the remainder of the time. For an *and*-combination we compute the proportion of the time that all incoming arcs are active. For an *or*-combination we compute the proportion of the time that at least one parent node is active. To compute the provability of the query given a document, we examine all paths leading from the document to the query and compute the proportion of time that all of the arcs on at least one proof path are active. Given the structure of these networks, this computation can be done using series parallel reduction of the subgraph joining the document and query in time proportional to the number of arcs in the subgraph.

In general, the Bayesian and Dempster-Shafer models are quite different and can lead to different results. Under the assumption of disjunctive rule interaction (so called "noisy-OR") and the interpretation of an arc from a to b as $P(b|a) = p$ and $P(b|\neg a) = 0$, the Bayesian and Demptser-Shafer models will produce similar results [27, page 446]. The document retrieval inference networks described here are based on the Bayesian inference network model.

The use of Bayesian inference networks for information retrieval represents an extension of probability-based retrieval research dating from the early 1960's [23]. It has long been recognized that some terms in a collection are more significant than others and that information about the distribution of

terms in a collection can be used to improve retrieval performance. The use of these networks generalizes existing probabilistic models and allows integration of several sources of knowledge in a single framework.

3 Basic Model

The basic document retrieval inference network, shown in figure 1, consists of two component networks: a document network and a query network. The document network represents the document collection using a variety of document representation schemes. The document network is built once for a given collection and its structure does not change during query processing. The query network consists of a single node which represents the user's information need and one or more query representations which express that information need. A query network is built for each information need and is modified during query processing as existing queries are refined or new queries are added in an attempt to better characterize the information need. The document and query networks are joined by links between representation concepts and query concepts. All nodes in the inference network are binary-valued and take on values from the set $\{false, true\}$.

3.1 Document network

The document network consists of document nodes (d_i's), text representation nodes (t_j's), and concept representation nodes (r_k's). Each document node represents an actual document in the collection. A document node corresponds to the event that a specific document has been observed. The form of the document represented depends on the collection and its intended use, but we will assume that a document is a well defined object and will focus on traditional document types (e.g., monographs, journal articles, office documents, ...).

Document nodes correspond to abstract documents rather than their physical representations. A text representation node or text node corresponds to a specific text representation of a document. A text node corresponds to the event that a text representation has been observed. We will focus here on traditional document texts, but one can easily imagine other content types for documents (e.g., figures) and multi-media documents might have several content representations (e.g., audio or video). In these cases, a single document might have multiple representations. Similarly, a single text content might be shared by more than one document. While this sharing is rare (an example would be a journal article that appears in both a serial issue and in a reprint collection) and is not generally represented in current retrieval models, it is common in hypertext systems. For clarity, we will only consider text representations and will assume a one-to-one correspondence between documents and texts. The dependence of a text upon the document is represented in the network by an arc from the document node to the text node.

The content representation nodes or representation nodes can be divided into several subsets, each corresponding to a single representation technique that has been applied to the document texts. For example, if a collection has been indexed using automatic phrase extraction and manually assigned index terms, then the set of representation nodes will consist of two distinct subsets or content representation types with disjoint domains. Thus, if the phrase "information retrieval" has been extracted and "information retrieval" has been manually assigned as an index term, then two representation nodes with distinct meanings will be created. One corresponds to the event that "information retrieval" has been automatically extracted from a subset of the collection, the second corresponds to the event that "information retrieval" has been manually assigned to a (presumably distinct) subset of the collection. We represent the assignment of a specific representation concept to a document by a directed arc to the representation node from each text node corresponding to a document to which the concept has been assigned. For now we assume that the presence or absence of a link corresponds to a binary assigned/not assigned distinction, that is, there are no partial or weighted assignments.

In principle, the number of representation schemes is unlimited. In addition to phrase extraction and manually assigned terms we would expect representations based on natural language processing and automatic keyword extraction. Refinements that can be applied to multiple representations (e.g., thesauri, term clustering, or inference rules) will be discussed

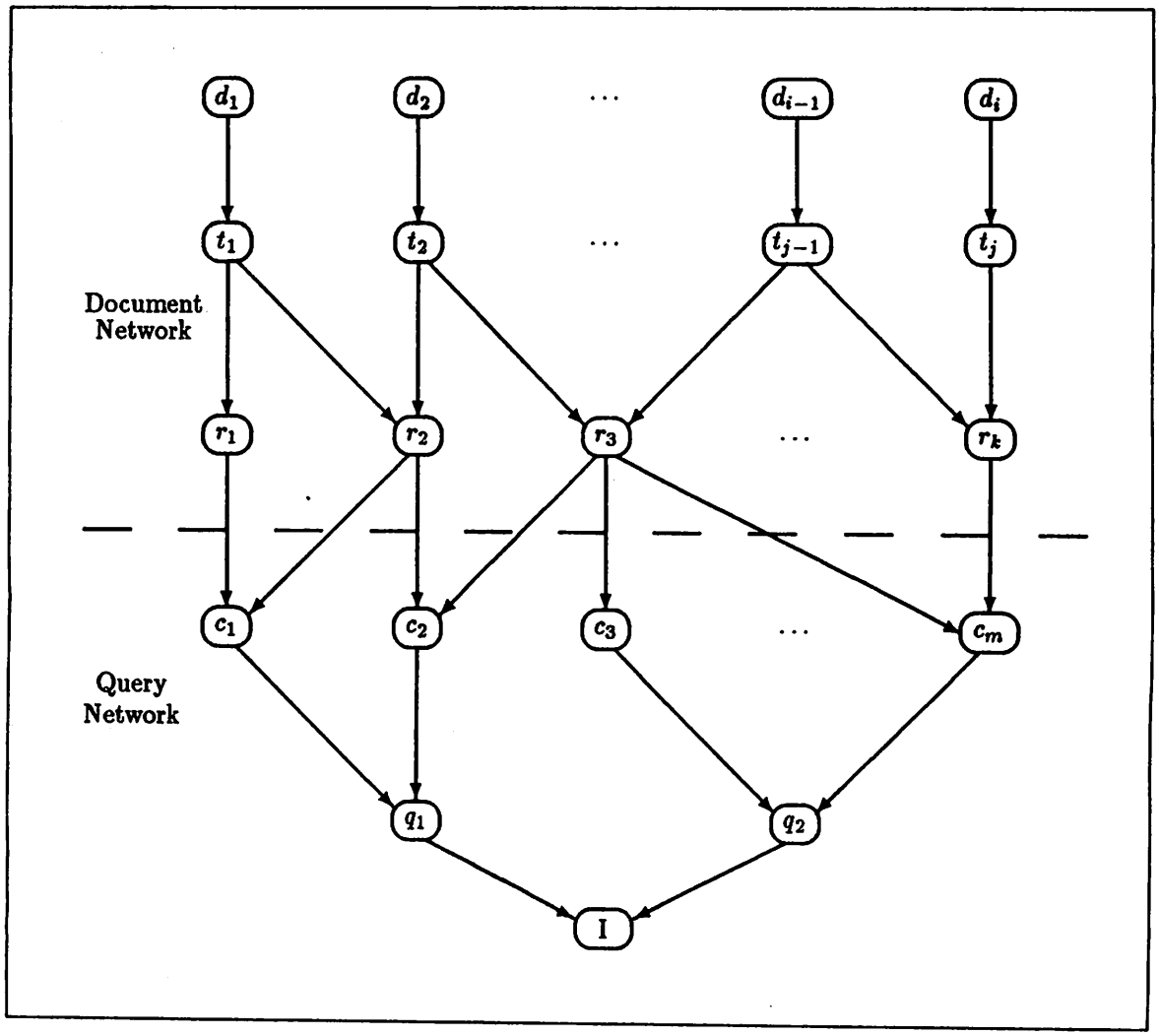

Figure 1: Basic document inference network

in section 5. For any real document collection, however, the number of representations used will be fixed and relatively small. The potential domain of each representation scheme may also be unlimited, but the actual number of primitive representation concepts defined for a given collection is fixed by the collection. The domain for most automated representation schemes is generally bounded by some function of the collection size (e.g., the number of keywords cannot exceed the number of words in a collection). For manual representation schemes the domain size is limited by the number of documents and the amount of time a human expert can invest to analyze each document.

The basic document network shown in figure 1 is a simple three level directed acyclic graph (DAG) in which document nodes are roots, text nodes are interior nodes, and representation nodes are leaves. Document nodes have exactly one text node as a child and each text node has one or more representation nodes as children.

Each document node has a prior probability associated with it that describes the probability of observing that document; this prior probability will generally be set to 1/(collection size) and will be small for reasonable collection sizes. Each text node contains a specification of its dependence upon its parent; by assumption, this dependence is complete, a text node is observed (t_i = *true*) exactly when its parent document is observed (d_i = *true*).

Each representation node contains a specification of the conditional probability associated with the

node given its set of parent text nodes. This specification incorporates the effect of any indexing weights (e.g., term frequency for each parent text) or term weights (e.g., inverse document frequency) associated with the representation concept. While, in principle, this would require $O(2^n)$ space for a node with n parents, in practice we will generally use canonical representations that will allow us to compute the required conditional probabilities when needed. These canonical schemes will usually require $O(n)$ space if we need to weight the contribution of each parent or $O(1)$ space if parents are to be treated uniformly.

3.2 Query network

The query network is an "inverted" DAG with a single leaf that corresponds to the event that an information need is met and multiple roots that correspond to the concepts that express the information need. As shown in figure 1, a set of intermediate query nodes may also be used in cases where multiple queries are used to express the information need. These nodes are a representation convenience; it is always possible to eliminate them by increasing the complexity of the distribution specified at the node representing the information need.

In general, the user's information need is internal to the user and is not precisely understood. We attempt to make the meaning of an information need explicit by expressing it in the form of one or more queries that have a formal interpretation. It is unlikely that any of these queries will correspond precisely to the information need, but some will better characterize the information need than others and several query specifications taken together may be a better representation than any of the individual queries.

The roots of the query network are query concepts, they correspond to the primitive concepts used to express the information need. A single query concept node may have several representation concept nodes as parents. A query concept node contains a specification of the probabilistic dependence of the query concept on its set of parent representation concepts. The query concept nodes define the mapping between the concepts used to represent the document collection and the concepts comprising the queries. In the simplest case, the query concepts are constrained to be the same as the representation concepts and each query concept has exactly one parent representation node. In a slightly more complex example, the query concept "information retrieval" may have as parents both the node corresponding to "information retrieval" as a phrase and the node corresponding to "information retrieval" as a manually assigned term. As we add new forms of content representation to the document network and allow the use of query concepts that do not explicitly appear in any document representation, the number of parents associated with a single query concept will tend to increase. In many ways, a query concept is similar to a representation concept that is derived from other representation concepts (see section 5.2 for a discussion of derived representation concepts) and in some cases it will be useful to "promote" a query concept to a representation concept. For example, suppose that a researcher is looking for information on a recently developed process that is unlikely to be explicitly identified in any existing representation scheme. The researcher is sufficiently motivated, however, to work with the retrieval system to describe how this new concept might be inferred from other representation concepts. If this new concept definition is of general interest, it can be added to the collection of representation concepts.

The attachment of the query concept nodes to the document network has no effect on the basic structure of the document network. None of the existing links need change and none of the conditional probability specifications stored in the nodes are modified.

A query node represents a distinct query form and corresponds to the event that the query is satisfied. Each query node contains a specification of the dependence of the query on the query concepts comprising it. The content of the link matrices that contain the conditional probabilities is discussed further in section 4, but it is worth noting that the form of the link matrix is largely determined by the type of query; a link matrix simulating a Boolean query is much different than a matrix simulating a probabilistic or weighted query.

The single leaf representing the information need corresponds to the event that an information need is met. In general, we cannot predict with certainty whether a user's information need will be met by an arbitrary document collection. The query network

is intended to capture the way in which meeting the user's information need depends on documents and their representations. Moreover, the query network is intended to allow us to combine information from multiple document representations and to combine queries of different types to form a single, formally justified estimate of the probability that the user's information need is met. If the inference network correctly characterizes the dependence of the information need on the collection, the computed probability provides a good estimate.

3.3 Use of the inference network

The inference network we have described is intended to capture all of the significant probabilistic dependencies among the variables represented by nodes in the document and query networks. Given the prior probabilities associated with the documents (roots) and the conditional probabilities associated with the interior nodes, we can compute the posterior probability or belief associated with each node in the network. Further, if the value of any variable represented in the network becomes known we can use the network to recompute the probabilities associated with all remaining nodes based on this "evidence."

The network, taken as a whole, represents the dependence of a user's information need on the documents in a collection where the dependence is mediated by document and query representations. When the query network is first built and attached to the document network we compute the belief associated with each node in the query network. The initial value at the node representing the information need is the probability that the information need is met given that no specific document in the collection has been observed and all documents are equally likely (or unlikely). If we now observe a single document d_i and attach evidence to the network asserting $d_i = true$ we can compute a new belief for every node in the network given $d_i = true$. In particular, we can compute the probability that the information need is met given that d_i has been observed in the collection. We can now remove this evidence and instead assert that some d_j, $i \neq j$ has been observed. By repeating this process we can compute the probability that the information need is met given each document in the collection and rank the documents accordingly.

In principle, we need not consider each document in isolation but could look for the subset of documents which produce the highest probability that the information need is met. While a general solution to this best-subset problem is intractable, in some cases good heuristic approximations are possible. Best-subset rankings have been considered in IR [35], and similar problems arise in pattern recognition, medical diagnosis, and truth-maintenance systems. See [27] for a discussion of the best-subset or belief revision problem in Bayesian networks. At present, we consider only documents in isolation since the approach is computationally simpler and because we would like to compare our results with earlier retrieval research that generally attempts to produce document rankings that are consistent with the Probability Ranking Principle [29] in which documents are considered in isolation.

The document network is built once for a given collection. Given one or more queries, we then build a query network that attempts to characterize the dependence of the information need on the collection. If the ranking produced by the initial query network is inadequate, we must add additional information to the query network or refine its structure to better characterize the meaning of the existing queries. This feedback process is quite similar to that used in current retrieval systems. The process of translating queries into network forms is discussed further in section 4.

3.4 Causation in Bayesian inference networks

The notion of causation, that one random variable can be perceived as causing another, is fundamental to Bayesian inference networks. By drawing an arc from node a to node b we are asserting that a in some sense causes b. If a is observed, then our belief in b is fixed by that observation (assuming b has no other parents). If we later observe b to have a value that conflicts with our computed belief we suspect that either the conditional probability $P(b|a)$ is incorrect or that the topology is wrong (either b has causes we haven't recognized or a does not, in fact, cause b). If, however, we first observe b then our belief in a changes because a is a potential explanation for

b, that is, the observation of b constitutes evidence confirming or disconfirming a.

While in many cases the direction of causation is clear (e.g., most instances of physical causation), in many others it is difficult to distinguish between causal and evidential support. For example, our network in figure 1 asserts that our belief in a set of query concepts causes our belief in the query that contains them. We could also have argued that our belief that the query is a representation of the information need causes our belief that the query concepts are useful. In this case we view the query concepts as evidence that supports our belief in the query.

In our network in figure 1 we assert that the observation of a document (or a set of documents) causes our belief in a text representation, which causes our belief in a set of representation concepts, which in turn cause belief in a set of query concepts, which cause our belief in a set of queries, which finally cause our belief that the documents support the information need. In fact, there are (at least) two other topologies that have some intuitive appeal. In the first, we simply invert the entire network. This structure asserts that the information need causes our belief in the queries, which cause our belief in the query concepts comprising them. While this chain of causation is at least plausible, the next step in which query concepts cause our belief in representation concepts, is not very appealing. Since documents and their representations have an existence independent of any query network, the query concepts cannot cause the representation concepts; our belief that a representation concept is assigned to a set of documents is not altered by the processing of a query.

A second variation, shown in figure 2 asserts that the documents are the ultimate causes of the document network and the information need is the ultimate cause of the query network. The two nets are connected by links establishing the dependence of query concepts on representation concepts. To see why this network does not capture our intuition about the relationships among variables we need to look more closely at how variables interact in these causal structures.

In figure 3a, two nodes a and b are potential causes of c. If c has not been observed, a and b are independent. Changes in our belief in a will affect our belief in c but will have no effect on our belief in b. This clearly does not model the desired behavior in figure 2 where we would like our beliefs about query concepts induced by the document network to propagate up the query network to affect our belief in the information need. If, in figure 3a we observe c, then a and b become dependent since they are both potential causes for c. In effect, they are competing explanations for c. If we now observe a to be true, our belief in b diminishes because a fully accounts for our observation of c. This behavior also fails to capture the desired relationship between the document and query networks in figure 2; we would expect that when belief in the representation concepts supporting the query concepts increases, our belief in the information need would also increase Clearly, the two networks are not competing to support the query concepts.

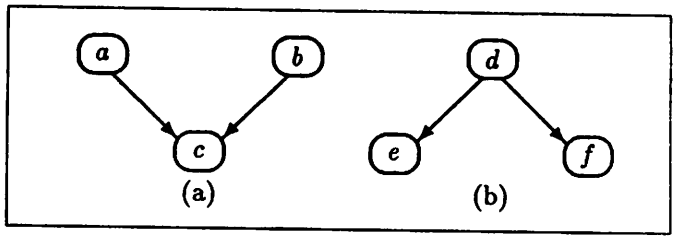

Figure 3: Basic causal topologies

Figure 3b shows the second case of interest. In this network a single node d causes both e and f. In the absence of any observation of d, e and f are dependent. If e is observed to be true, this evidence raises belief in d which in turn raises belief in f. In the absence of any evidence at d, any evidence collected at one of its children directly affects all children. Once d is observed, however, e and f become independent; further evidence gathered at one child will not affect belief in d (since its value is known) and can therefore have no impact on its siblings.

The interaction between the variables in these two network fragments helps to clarify the nature of "causation" in Bayesian inference nets. Two causes sharing a common consequence are independent until the consequence is observed, thereafter they compete. Two consequences of a common cause are dependent and share support until the cause is ob-

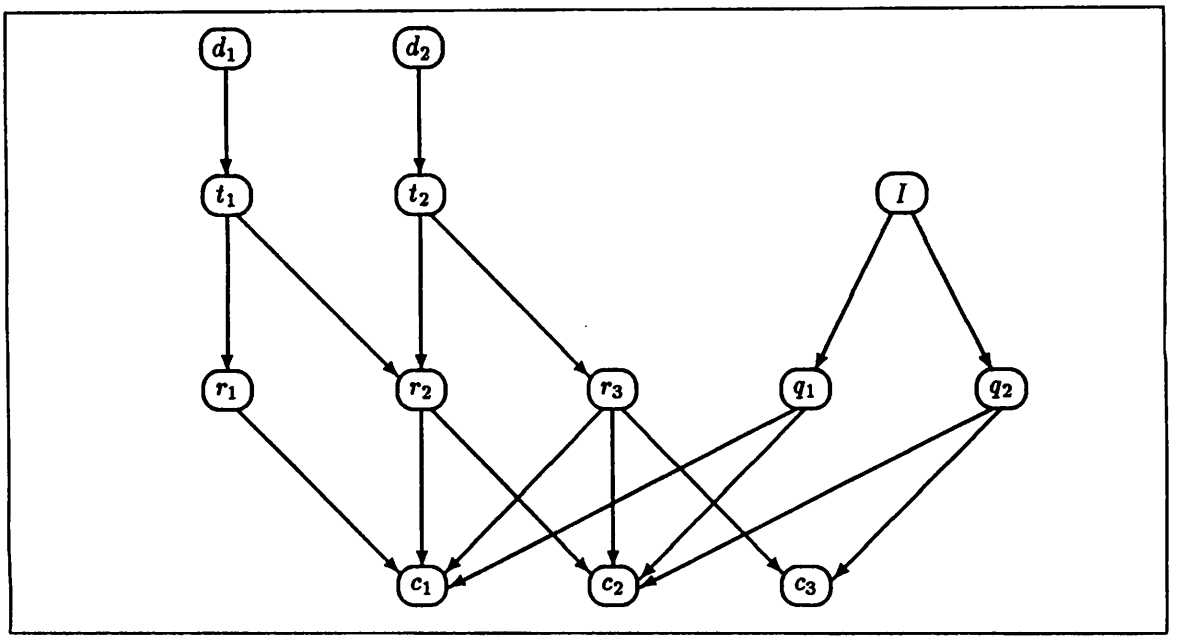

Figure 2: Mixed document inference network

served. Thereafter, the consequences are independent.

4 Comparison with other retrieval models

The inference network retrieval model bears similarities with both probabilistic and Boolean retrieval models. The inference networks can be used to simulate both probabilistic and Boolean queries and can be used to combine results from multiple queries.

4.1 Probabilistic retrieval models

Conventional probabilistic models [35,1,29] rank documents by the probability of each document's relevance to a given query, $P(\text{relevant}|d_i)$.[1] This is, in many ways, similar to computing the probability that a user's information need is met given a specific document, $P(I|d_i)$. The principal differences between conventional probabilistic models and the model described here are: 1) conventional probabilistic models do not explicitly represent the query, 2) conventional probabilistic models do not distinguish between a document and its representations but treat a document as a single vector, and 3) the inference model depends less upon Bayesian inversion than probabilistic models, Bayesian inversion is just one component of the procedure to estimate $P(I|d_i)$.

[1] Most probabilistic models do not actually compute $P(\text{relevant}|d_i)$, but simply rank documents using some function that is proportional to $P(\text{relevant}|d_i)$. Like Fuhr ([15]), we believe that an estimate of the probability of relevance is more useful than the ranking by itself. A ranked list of documents in which the top ranked document has a probability of relevance of 0.5 should be viewed differently than a similar list in which the top ranked document has a probability of relevance of 0.95.

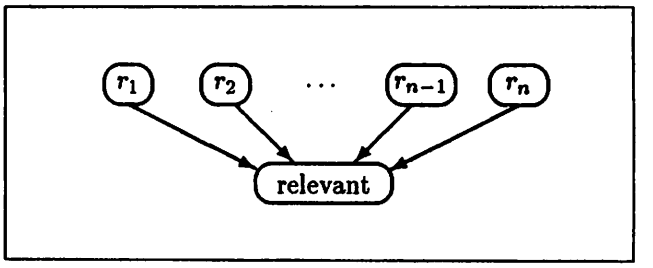

Figure 4: Network for binary independence model

An inference network that corresponds to the binary independence model [35,14] is shown in figure 4. A document is represented by a vector whose components are indexing or representation concepts ($d_i = \{r_1, \ldots, r_n\}$). The set of concepts considered is generally restricted to the subset that actually occur in the query. Comparing this network with that shown in figure 1, we see that in

the binary independence model, the document network is represented by a single level of representation nodes and the query network consists of a single relevance node. In order to implement this network we must somehow estimate the probability of relevance given the set of parent representation concepts and this estimate must incorporate all of our judgments about the probability that a representation concept should be assigned to a document, about the semantic and stochastic relationships between representation concepts, about the relationship between concepts named in the query and assigned to documents, and about the semantics of the query itself. This dependence is complex and its estimation is not a task we could expect users to perform willingly or reliably.

One approach to simplifying the estimation task is to invoke Bayes' rule so that we need only estimate the probability that each representation concept occurs in relevant or non-relevant documents. This approach does not help to provide initial estimates of the probability distributions since these "simpler" estimates must still incorporate all of the judgments required for the "hard" estimate. The advantage of this approach is that, given samples of relevant and non-relevant documents, it is easy to compute $P(r_i)$ for the relevant sample and to use the result as an estimate of $P(r_i|\text{relevant} = \textit{true})$ and similarly for $P(r_i|\text{relevant} = \textit{false})$. Given a set of independence assumptions and estimates for $P(d_i)$ and $P(\text{relevant} = \textit{true})$ we can compute $P(\text{relevant}|d_i)$.[2] Estimating $P(\text{relevant}|d_i)$ without the use of Bayes' rule would be extremely difficult [22].

Essentially the same procedures can be used to estimate $P(Q|d_i)$. The main difference between to two estimates is that instead of using the representation concepts directly we must compute $P(c_j|\pi_{c_j})$ and compute an expected value for $P(c_j|d_i)$ in order to estimate $P(Q|d_i)$.

The question remains, however, whether estimates of $P(\text{relevant}|d_i)$ or $P(Q|d_i)$ obtained in this way match users' intuition about the dependence. The fact that relevance feedback does improve retrieval performance suggests that the estimates of $P(\text{relevant}|d_i)$ do capture at least some of the dependence, but these estimates are generally based on a small number of relevant documents and are necessarily rather coarse.

While it is clear that estimating $P(\text{relevant}|d_i)$ directly is impractical, it may be possible to obtain estimates of $P(Q|\pi_Q)$. Users may, for example, be able to assign importance to the concepts in their query and may be able to identify significant interactions between concepts. These estimates could improve the initial estimate and might be used in conjunction with the estimates derived from training samples.

A second approach to simplifying the estimation task is to identify the different types of judgments that enter into the overall estimate and to develop estimates for each type of judgment separately. The model presented here represents one decomposition in which the task of estimating the probability that a given document satisfies an information need consists of judgments about the relationship of a document to its text, the assignment of representation concepts to the text, the relationships between query and representation concepts, and the relationship between queries, query concepts, and the information need. Other decompositions are certainly possible and can be accommodated within the same general framework. The set of relationships presented here incorporates those judgments most important for current generation document retrieval systems.

When viewed this way, the probabilistic and inference models use two similar approaches to the same estimation problem. The probabilistic model uses a single, general purpose rule and makes assumptions about term dependence in order to estimate $P(\text{relevant}|d_i)$. The model presented here views the problem of estimating $P(I|d_i)$ as consisting of a set of logically related estimates. Each estimate is made independently using procedures specific to the type of estimate; the "probabilistic" estimate of $P(Q|\pi_Q)$ is simply one component of the overall estimate. The component estimates are then combined in a manner consistent with the dependence relationships represented in the inference network to provide an estimate of $P(I|d_i)$.

4.2 Boolean retrieval

For all non-root nodes in the inference network we must estimate the probability that the node takes on

[2] $P(d_i)$ and $P(\text{relevant} = \textit{true})$ do not play a major role in probabilistic models that only produce a document ranking but are required to compute $P(\text{relevant}|d_i)$.

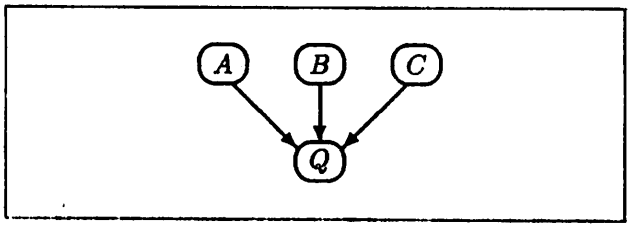

Figure 5: Network for link matrix examples

a value given any set of values for its parent nodes. The most direct way to encode our estimate is a matrix (a *link* matrix). Since we are dealing with binary valued propositions, this matrix specifies the probability that a node a takes the value *true* or *false* for all combinations of parent values. The update procedures for Bayesian networks then use the likelihood information provided by the set of parents to condition over the link matrix values to compute our belief in a or $P(a = false, a = true)$. Similarly, the link matrix is used to provide diagnostic information to the set of parents based on our belief in a. As discussed earlier, encoding our estimates in link matrix form is practical only for nodes with a small set of parents, so our estimation task has two parts: how do we estimate the dependence of a node on its parents and how do we encode these estimates in a usable form?

In order to simulate Boolean retrieval, we use three canonical link matrix forms that implement the logic operations *and*, *or*, and *not*. A fourth common link matrix form implements a weighted product. For illustration, we will assume that a node Q has three parents A, B, and C (figure 5) and that

$$\begin{aligned} P(A = \text{true}) &= a \\ P(B = \text{true}) &= b \\ P(C = \text{true}) &= c. \end{aligned}$$

By a canonical form we mean that, given an ordering on a set of n parents, we can compute the link matrix values $L[i,j]$, $i \in \{0,1\}, 0 \leq j < 2^n$ given the set of parent indices corresponding to j.

For *or*-combinations, Q will be true when any of A, B, or C is true. This suggests a link matrix of the form

$$L_{or} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Using a closed form of the update procedures gives

$$P(Q = true) = a + b + c - (ab + bc + ac) + abc$$

which is the familiar rule for disjunctive combination of events.

For *and*-combinations, Q is true only when A, B, and C are all true and we have a matrix of the form

$$L_{and} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Again, the closed form update procedure gives

$$P(Q = true) = abc$$

which is the familiar rule for conjunctive combination of events.

The *not* operator is defined only for unary propositions. If Q has the parent A, Q = *true* exactly when A = *false* which suggests a link matrix of the form

$$L_{not} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Again using a closed form we have

$$P(Q = true) = 1 - a.$$

If we restrict the parent nodes for any of the logic operators to values 0 or 1 then the inference networks can be used to evaluate arbitrary Boolean expressions and can simulate Boolean retrieval with binary indexing. If we allow terms to take on weights in the range $[0,1]$ and interpret these weights as the probability that the term has been assigned to a document text, then these inference networks provide a natural interpretation for Boolean retrieval with weighted indexing. Moreover, both prababilistic and Boolean query forms can be attached to the same document network and their results can be combined in the estimate of $P(I|d_i)$.

A fourth link matrix form arises when our belief that Q is true depends only on the number of parents that are true. If j corresponds to a link matrix column for which m parents are true, then

$$\begin{aligned} L_{sum}[1,j] &= \frac{m}{n} \\ L_{sum}[0,j] &= \frac{n-m}{n}. \end{aligned}$$

For our three parent example

$$L_{sum} = \begin{pmatrix} 1 & \frac{2}{3} & \frac{2}{3} & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{2}{3} & \frac{1}{3} & \frac{2}{3} & \frac{2}{3} & 1 \end{pmatrix}.$$

Evaluation of this link matrix form results in

$$P(Q = true) = \frac{a+b+c}{3}.$$

In this matrix form all parents are weighted equally, but a number of other weightings are possible. For example, we can choose weights so that the first parent observed has the most influence on our belief in Q (essentially, the *or*-combination is an extreme case) or we can choose weights so that the first parent observed has little or no influence and the second and third parents determine our belief.

4.3 Estimating the probabilities

Given the link matrix forms of the last section, we now consider the estimates required for the basic model shown in figure 1. The only roots in the inference network of figure 1 are the document nodes and the prior probabilities associated with these nodes is set to 1/(collection size). Estimates are required for five different node types: text, representation and query concepts, query, and information need.

Text nodes. Since text nodes are completely dependent upon the parent document node, the estimate is straightforward. Since there is a single parent, a matrix form can be used; t_i is true exactly when d_i is true and false exactly when d_i is false so

$$L_{text} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

This matrix form is the inverse of that used for *not*. If we allow document nodes to share text nodes, then an *or* matrix is appropriate, t_i is true when any parent is instantiated.

Representation concept nodes. As suggested in section 3.1, the belief in a representation concept depends both on how strongly that term is supported by any parent document texts (indexing weights) and on the quality of the term (term weights). A variety of techniques have been developed for estimating these indexing and term weights [35]. Any term and indexing weight schemes can be used in the network provided that the weights can be scaled to the range [0, 1] and interpreted as the probability that the given representation concept is observed given the set of parent text nodes that have been instantiated (in the basic model only one parent node can be instantiated and that node must be a text node. These restrictions are relaxed in the extended model of section 5).

For binary indexing and unweighted terms an *or*-combination can be used so that $P(r_i = true) = 1$ if any parent is instantiated. For term weights, we can use a variation on the *or*-combination so that

$$P(r_i = true) = \text{term weight}$$

if any parent is instantiated. For indexing weights, it is necessary to store the indexing weight for each parent in the representation concept node. If only one parent can be instantiated we can set

$$P(r_i = true) = \text{weight for instantiated parent.}$$

If multiple parents can be instantiated then we also need a rule to combine their effect. Finally, for weighted terms and indexing we need to combine the effect of the two weights to form the estimate.

Query concept nodes. Much of the research on indexing and term weights can be incorporated into the document network. The query network, particularly the links between representation and query concepts is less well understood. We are interested in estimating the probabilistic dependence of concepts mentioned in the user's query upon the representation concepts. Most current retrieval models view these two sets of concepts as equivalent under the assumption that the user knows the set of representation concepts and can formulate queries using the representation concepts directly. Research suggests, however, that the mismatch between query and indexing vocabularies may be a major cause of poor recall [16]. While our initial implementation will be limited to linking query concepts to "nearly" equivalent representation concepts using a weighted product combination rule, it would appear that improved estimates of the dependence of query concepts on representation concepts could markedly improve performance. Two areas of research bear directly on improving the quality of these estimates:

automatic thesaurus construction and natural language research aimed at extracting concept descriptions from query text, identifying synonymous or related descriptions, and resolving ambiguity [19].

Query nodes. The dependence of query nodes on the query concepts is more straightforward. For Boolean queries we can build an evaluation tree using the link matrix forms described in section 4.2; we can adjust link matrix values if we have information about the relative importance of the query concepts. For probabilistic queries we can use the estimates described in section 4.1.

Information need. The information need will generally be expressed as a small number of queries of different types (Boolean, probabilistic, natural language). These can be combined using a weighted-product link matrix with weights adjusted to reflect any user judgments about the importance or completeness of the individual queries.

5 Extensions to the basic model

The basic model described in section 3 is limited in at least two respects. First, we have assumed that evidence about a variable establishes its value with certainty. Second, we have represented only a limited number of dependencies between variables. In this section we will see that these limitations can be removed.

5.1 Uncertain evidence and feedback

The only use of evidence in the basic model is to assert that a document representation has been observed (d_i = *true*). During query processing we assert each document true and rank documents based on the probability that the information need is met. Since we do not assert that the remaining documents are *false*, they continue to contribute to the belief that the information need is met so that, while we instantiate documents in isolation, the resulting probability is dependent upon both the instantiated document and some subset of the uninstantiated documents in the collection. In real document collections, the prior probability associated with each document is small and only a small portion of the representation concepts will bear on the information need, so the contribution of these uninstantiated documents will generally be small compared to the contribution of the instantiated document.

Evidence is attached to a node a in a Bayesian network by creating a new *evidence* node b as a child of a. This new node b then passes a likelihood vector (both components of a likelihood ratio) to a. The evidential support for a is then the product of the likelihood vectors from b and any other children. Since evidence is expressed in terms of likelihood we are not restricted to the values *true* and *false* (the vectors $(0,1)$ and $(1,0)$, respectively) but need only specify the likelihood of a = *true* and a = *false* given the evidence summarized at b. As a result, evidence can be used as a weight associated with a node. For example, if we attach confirming (disconfirming) evidence to a representation node it raises (lowers) the belief in all documents containing it, in all query concepts and queries that use it, and in the information need. The effect of the evidence is to bias the node so that the positive (negative) belief component passed to parents or children is amplified and the negative (positive) component is attenuated. If the evidence entirely confirms or disconfirms the node, then it blocks the flow of belief/evidence entirely – essentially it infinitely amplifies one component and attenuates the other so that the belief/evidence passed on is independent of the support received from parents or other children.

A side effect of using evidence in this way is that it establishes a coupling between documents containing the representation concept. When we instantiate one document, belief in all other documents containing the same representation concept will be reduced. This effect is probably not significant for a single term, but if two documents had similar indexing and all common terms had evidence attached, the coupling could be pronounced.

One potential use for this kind of weight is to implement a form of feedback. If, as a result of relevance feedback, a query, query concept, or representation concept is found to be more or less important than others, its effect on the propagation of belief through the network can be altered by attaching evidence. Frisse and Cousins [13] use this approach to implement feedback in a hierarchy of index terms associated with a hypertext medical handbook.

In principle, the link matrix associated with a representation concept contains the probability that

that concept is true given any set of parent beliefs. This probability depends both on the descriptive quality of the term and on the specific parent documents that are instantiated. In practice, we cannot store the matrix for nodes that have more than a few parents. Instead, we store the indexing weight associated with each parent, the term weight associated with the representation concept, and a function that computes the desired probability based on these weights.

In some cases we can manipulate the function used to compute the conditional probability in order to adapt the behavior of the network. This approach could be used as an alternative to evidence when implementing feedback. The two approaches are fundamentally different. When using evidence, the original probability distribution defined by the network is always maintained. Manipulating the combining function will generally alter the probability distribution. Manipulating the combining function will not generally be useful in the document network where the distribution models the statistical and semantic relations in the collection and its representation, but it may be useful in the query network where the dependence relations are much less constrained. The document network is largely fixed by the collection and our choice of representations; during query processing we attempt to build a network that "correctly" characterizes the dependence of the information need on that collection.

5.2 Additional dependencies

In the basic model, we assume that there are no dependencies between documents, between texts, between representation concepts, between query concepts, or between queries. While independence assumptions like these are not uncommon in retrieval models, it is widely recognized that the assumptions are unrealistic; there are a number of both statistical and logical dependencies between representation concepts and between documents. In particular, we would like to incorporate term and document clustering and would like to represent citation links between documents and thesaurus relationships between terms.

The basic mechanism for representing these dependencies is unchanged, we identify the set of nodes upon which a given node depends and characterize the probability associated with each node conditioned on its immediate parents. When adding these new links, however, we must be careful to preserve the acyclic nature of the inference network. Bayesian inference networks cannot represent cyclic dependencies, in effect evidence attached to any node in the cycle would continually propagate through the network and repeatedly reinforce the original node. In the basic model, no cycles are possible since nodes are only linked to node types that are lower in the DAG. The introduction of these "horizontal" dependencies makes cycles possible.

Document and term clustering. A variety of clustering techniques have been developed for information retrieval [35]. These may be loosely categorized as *document* clustering techniques which attempt to divide the collection into (possibly overlapping) subsets which are similar and *term* clustering techniques which attempt to identify subsets of representation concepts with similar usage or meaning. Clustering techniques differ widely in the document or term attributes considered, the definition of a similarity or dissimilarity measure, and the structure of the resulting classification. Term clustering techniques represent one kind of automatically-built thesaurus in which terms contained in a cluster are, in some sense, synonymous; clusters may be organized in a hierarchy to represent broader and narrower classifications. Representation of these thesaurus-like relationships will be discussed shortly.

Document clustering techniques are generally used to find documents that are similar to a document that is believed relevant under the assumption that similar documents are related to the same queries. Our use of cluster information is somewhat different since we do not retrieve clusters, but we can incorporate the cluster information in the dependence relationships between document texts and representation concepts. In the fragment shown in figure 6, document texts t_1, t_2, and t_3 are indexed using representation concepts r_1, r_2, r_3, and r_4. Documents t_2 and t_3 have been identified as part of cluster c_1; both texts are linked to a cluster node and the cluster node is linked to the representation concepts that define the cluster. The cluster node is similar to a conventional cluster representative. Documents t_1 and t_2 are indexed by the same representation concepts

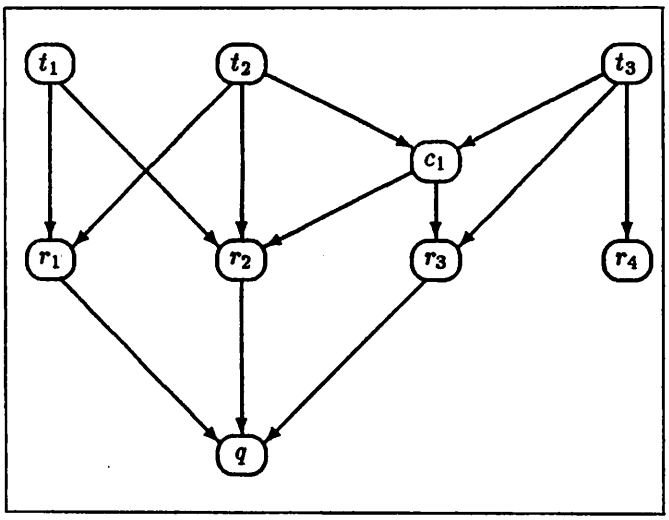

Figure 6: Document clustering model

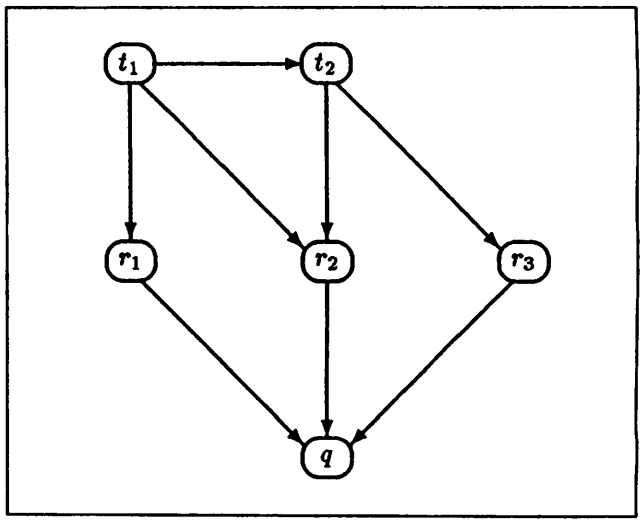

Figure 7: Nearest neighbor link

(r_1 and r_2) and, if we assume equivalent conditional probabilities, would be ranked equivalently in the absence of the cluster node. With the addition of the cluster node, however, a new representation concept (r_3) is associated with t_2 by virtue of its cluster membership. Assuming that r_3 contributes positively to the belief in q, t_2 would be ranked higher than t_1. Like query nodes, cluster nodes are a representation convenience, it is always possible to eliminate them by increasing the complexity of the distribution specified at the representation concept nodes.

Citation and nearest neighbor links. A variety of asymmetric relationships between pairs of documents can also be represented. These relationships are similar to clustering in that they use an assumed similarity between documents to expand the set of representation concepts that can be plausibly associated with a text. They differ in that they are ordered relations defined on pairs of documents rather than an unordered, set membership relationship between documents and clusters.

Perhaps the best example of this kind of relationship is the nearest neighbor link in which a document is linked to the document judged to be most similar to the original document. In figure 7 the set of representation concepts associated with document t_1 is expanded by virtue of its nearest neighbor link to document t_2. Note that it is not possible to simultaneously represent t_2 as t_1's nearest neighbor and t_1 as t_2's nearest neighbor since the pair of links would induce a cycle. A second kind of ordered link is based on citations occurring in the text. Citation links may be useful if the type of reference can be determined (e.g., citing a similar work, a peripherally related work, or a work presenting an opposing viewpoint) to allow estimation of the probabilistic dependence between the nodes.

Thesaurus relationships. The structure of these networks provides a natural mechanism to represent probabilistic dependencies between the concepts or terms that describe documents and information needs. These relationships are similar to conventional thesaurus relationships, but include more information. For example, a conventional thesaurus might list "house pet" as a broader term for "dog" and "cat"; the network representation will include a specification of the probability that "house pet" should be assigned given a document containing "dog" or "cat" in isolation, neither term, or both terms.

Synonyms, related terms, and broader terms can be represented by creating new nodes to represent the synonym or related term class or the broader term and adding the new node as a child to the relevant representation concept node. We will generally prefer to add these nodes as part of the query network since their presence in the document network would represent a computational burden even when not used in a query. Although generally less useful, narrower term relationships can also be represented.

6 Conclusion

The retrieval model presented here provides a framework within which to integrate several document representations and search strategies. We are currently refining the model and are planning experiments to compare search performance based on this model with that of other models and to compare performance of potential representations and search strategies.

Acknowledgments

This work was supported in part by OCLC Online Computer Library Center, by Rome Air Development Center and Air Force Office of Scientific Research under contract F30602-85-C-0008, and by NSF Grant IRI-8814790.

References

[1] N. J. Belkin and W. B. Croft. Retrieval techniques. In M. E. Williams, editor, *Annual Review of Information Science and Technology*, chapter 4, pages 109–145. Elsevier Science Publishers, 1987.

[2] P. Cheeseman. An inquiry into computer understanding. *Computational Intelligence*, 4:58–66, Feb. 1988. Article is part of a debate between logic and probability schools in AI.

[3] P. R. Cohen. *Heuristic Reasoning About Uncertainty: An Artificial Intelligence Approach.* Pitman, Boston, MA, 1985.

[4] P. R. Cohen and R. Kjeldsen. Information retrieval by constrained spreading activation in semantic networks. *Information Processing and Management*, 23(2):255–268, 1987.

[5] W. S. Cooper. A definition of relevance for information retrieval. *Information Storage and Retrieval*, 7:19–37, 1971.

[6] W. B. Croft. A model of cluster searching based on classification. *Information Systems*, 5:189–195, 1980.

[7] W. B. Croft. Approaches to intelligent information retrieval. *Information Processing and Management*, 23(4):249–254, 1987.

[8] W. B. Croft and R. H. Thompson. I^3R: A new approach to the design of document retrieval systems. *Journal of the American Society for Information Science*, 38(6):389–404, Nov. 1987.

[9] W. B. Croft and H. Turtle. A retrieval model incorporating hypertext links. In *Hypertext '89 Proceedings*, pages 213–224, 1989.

[10] A. P. Dempster. A generalization of Bayesian inference. *Journal of the Royal Statistical Society B*, 30:205–247, 1968.

[11] J. Doyle. A truth maintenance system. *Artificial Intelligence*, 12(3):231–272, 1979.

[12] J. Fox. Three arguments for extending the framework of probability. In L. N. Kanal and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 447–458. North-Holland, Amsterdam, 1986.

[13] M. E. Frisse and S. B. Cousins. Information retrieval from hypertext: Update on the dynamic medical handbook project. In *Hypertext '89 Proceedings*, pages 199–212, 1989.

[14] N. Fuhr. Models for retrieval with probabilistic indexing. *Information Processing and Management*, 25(1):55–72, 1989.

[15] N. Fuhr. Optimum polynomial retrieval functions based on the probability ranking principle. *ACM Transactions on Information Systems*, 7(3):183–204, July 1989.

[16] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Commun. ACM*, 30(11):964–971, Nov. 1987.

[17] L. N. Kanal and J. F. Lemmer, editors. *Uncertainty in Artificial Intelligence.* North-Holland, Amsterdam, 1986.

[18] J. Katzer, M. J. McGill, J. A. Tessier, W. Frakes, and P. DasGupta. A study of the overlap among document representations. *Information Technology: Research and Development*, 1:261–274, 1982.

[19] R. Krovetz and W. B. Croft. Word sense disambiguation using a machine readable dictionary. In *Proceedings of the 12th International Conference on Research and Development in Information Retrieval*, pages 127–136, 1989.

[20] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society B*, 50(2):157–224, 1988.

[21] J. F. Lemmer and L. N. Kanal, editors. *Uncertainty in Artificial Intelligence 2.* North-Holland, Amsterdam, 1988.

[22] D. Lewis, W. B. Croft, and N. Bhandaru. Language-oriented information retrieval. *International Journal of Intelligent Systems*, 1989. Forthcoming.

[23] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *J. ACM*, 7:216–244, 1960.

[24] M. McGill, M. Koll, and T. Noreault. An evaluation of factors affecting document ranking by information retrieval systems. Technical report, Syracuse University, School of Information Studies, 1979. Funded under NSF-IST-78-10454.

[25] N. J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28(1):71–87, 1986.

[26] R. N. Oddy, R. A. Palmquist, and M. A. Crawford. Representation of anomalous states of knowledge in information retrieval. In *Proceedings of the 1986 ASIS Annual Conference*, pages 248–254, 1986.

[27] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, 1988.

[28] S. E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33(4):294–304, Dec. 1977.

[29] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[30] G. Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, 1976.

[31] G. Shafer. Belief functions and possibility measures. In J. Bezdek, editor, *Analysis of Fuzzy Information, Volume 1: Mathematics and Logic*, pages 51–58. CRC Press, Boca Raton, FL, 1987.

[32] K. H. Stirling. The effect of document ranking on retrieval system performance: A search for an optimal ranking rule. *Proceedings of the American Society for Information Science*, 12:105–106, 1975.

[33] R. H. Thompson and W. B. Croft. Support for browsing in an intelligent text retrieval system. *International Journal of Man-Machine Studies*, 30:639–668, 1989.

[34] R. M. Tong and D. Shapiro. Experimental investigations of uncertainty in a rule-based system for information retrieval. *International Journal of Man-Machine Studies*, 22:265–282, 1985.

[35] C. J. van Rijsbergen. *Information Retrieval.* Butterworths, 1979.

[36] C. J. van Rijsbergen. A non-classical logic for information retrieval. *Computer Journal*, 29(6):481–485, 1986.

[37] P. Wilson. Situational relevance. *Information Storage and Retrieval*, 9:457–471, 1973.

[38] L. A. Zadeh. The role of fuzzy logic in the management of uncertainty in expert systems. *Fuzzy Sets and Systems*, 11:199–228, 1983.

[39] L. A. Zadeh. Is probability theory sufficient for dealing with uncertainty in AI: A negative view. In L. N. Kanal and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 103–116. North-Holland, Amsterdam, 1986.

[40] L. A. Zadeh. A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination. *AI Magazine*, 7(2):81–90, 1986.