Inference Networks for Document Retrieval

Howard Turtle and W. Bruce Croft
Computer and Information Science Department
University of Massachusetts
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 [TC89], document clustering [Cro80], spreading activation search [CK87], support for multiple search strategies [CT87], and representation of user knowledge [OPC86] or document content [TS85].

Recent work suggests that significant improvements in retrieval performance will require techniques that, in some sense, "understand" the content of documents and queries [vR86, Cro87] 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 mechanisms for inferring these kinds of relationships [CT89,CK87].

The idea that retrieval is an inference or evidential reasoning process is not new. Cooper's logical relevance [Coo71] is based on deductive relationships between representations of documents and information needs. Wilson's situational relevance [Wil73] 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 similar to those used in expert systems that must reason with uncertain information. A number of competing inference models have been developed for these kinds of expert systems [KL86,LK88] 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 repre-

sentations, even when the average retrieval performance achieved with each representation is the same. Katzer, for example, found little overlap in documents retrieved using seven different representations, but found that documents retrieved by multiple representations were likely to be relevant [KMT+82]. Similar results have been obtained when comparing term- with cluster-based representations [CH79] and term- with citation-based representations [FNL88].

- Allow results from different queries and query types to be combined. Given a single natural language description of an information need, different searchers will formulate different queries to represent that need and will retrieve different documents, even when average performance is the same for each searcher [MKN79,KMT+82]. Again, documents retrieved by multiple searchers are more likely to be relevant. A description of an information need can be used to generate several query representations (e.g., probabilistic, Boolean), each using a different query strategy and each capturing different aspects of the information need. These different search strategies are known to retrieve different documents for the same underlying information need [Cro87].

- 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 [FLGD87]. Recall can be improved using domain knowledge to match query and representation concepts without significantly degrading precision.

The resulting formal retrieval model integrates several previous models in a single theoretical framework; multiple document and query representations are treated as evidence which is combined to estimate the probability that a document satisfies a user's information need.

In what follows we briefly review candidate inference models, present an inference network-based retrieval model, and compare the network model to current retrieval models.

2 Inference networks

The development of automated inference techniques that accommodate uncertainty has been an area of active research in the artificial intelligence community, particularly in the context of expert systems [KL86,LK88]. Popular approaches include those based on purely symbolic reasoning [Coh85,Doy79], fuzzy sets [Zad83], and a variety of probability models [Nil86,Che88]. Two inference models based on probabilistic methods are of particular interest: Bayesian inference networks [Pea88,LS88] and the Dempster-Shafer theory of evidence [Dem68,Sha76].

A Bayesian inference network is a directed, acyclic dependency graph (DAG) 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 *link* matrix that specifies $P(q|p)$ for all possible values of the two variables. When a node has multiple parents, the link 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.[1] 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.

The Dempster-Shafer theory of evidence, although not originally cast as a network model, 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.

The Bayesian and Dempster-Shafer models are different and can lead to different results. However, 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 Dempster-Shafer models will produce similar results [Pea88, 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 [MK60]. 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.

[1] While this probability specification is generally referred to as a link matrix, it is actually a tensor.

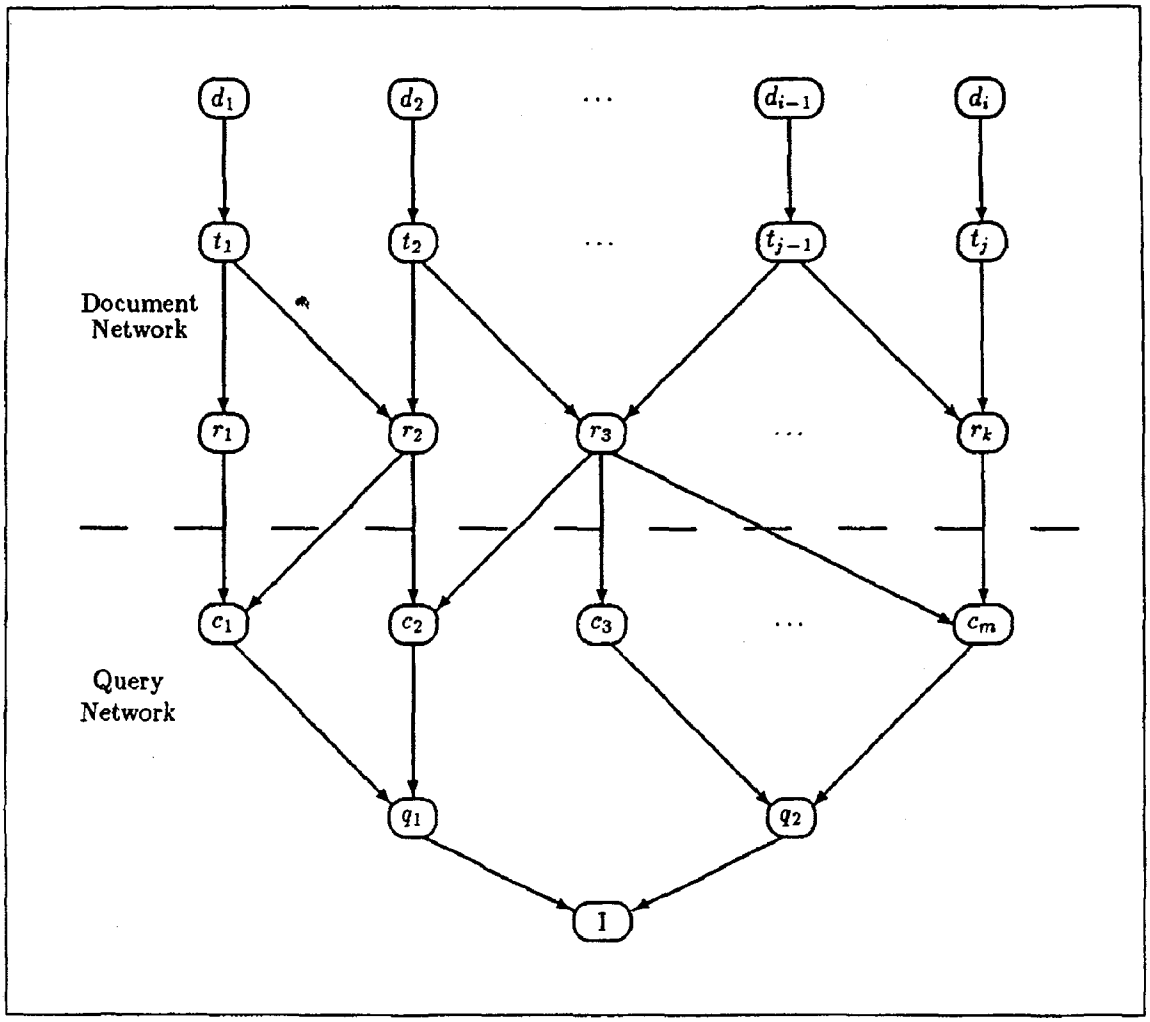

Figure 1: Basic document inference network

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 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 a 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 focus here on the text content of documents, but the network model can support documents nodes with multiple children representing additional component types (e.g., figures, audio, or video). Similarly, a single text might be shared by more than one document. While shared components is rare in traditional collections (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 consider only 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. 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

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.