ReferralWeb: A Resource Location System Guided by Personal Relations

by

Mehul A. Shah

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

May 1997

©1997 Mehul A. Shah. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis and to grant others the right to do so.

Signature of Author .....
Department of Electrical Engineering and Computer Science
May 29, 1997

Certified by .....
David R. Karger
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

ReferralWeb: A Resource Location System Guided by Personal Relations

by

Mehul A. Shah

Submitted to the Department of Electrical Engineering and Computer Science
on May 29, 1997, in partial fulfillment of the
requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

We describe the design and implementation of ReferralWeb, a system for identifying experts on keyword queries and generating a path of social relations by which to contact them. This system models and extracts existing social and professional relationships in the computer science community by mining publicly available documents on the internet. Using similar techniques, experts are also isolated from indexed web documents. A user interface combines the reconstructed social network and search engines to allow exploration and visualization of one's local personal network. We describe interviews and experiments which indicate that the current prototype fulfills a need not addressed by other public services. Finally, possible solutions for improved robustness and further evolution are proposed.

Thesis Supervisor: David R. Karger
Title: Assistant Professor, Dept. of Electrical Engineering and Computer Science

Acknowledgments

This thesis is dedicated to my beloved grandfather, the late Nandlal L. Shah. This is not the end but just the begining.

I want to thank my parents, sister, and my fiance Nupur Gupta for their neverending support. I want to thank Jamie Cho for looking around for experts. I want to thank Anne Hunter who made sure I would graduate. Finally, I would like to thank Prof. David R. Karger for his endless patience, invaluable advice, and painstaking scrutiny. This thesis would be *even* more confusing without his influence.

I also want to thank Henry Kautz and Bart Selman for allowing me to work on this topic and for providing support to a poor graduate student. This work was inspired by their ideas and guided and developed jointly with their expertise.

Contents

Chapter 1

Introduction

The recent large-scale internetworking of computers provides a medium for personal communication, collaboration, and the dissemination of information. The enormous volume and variety of information online has precipitated the need and development of tools, such as searchable web indexes and recommender systems, for browsing and locating pertinent resources. However, tools for discovering people who can provide specific resources, tangible or intangible, are still in their infancy.

Locating people online traditionally involves searching a directory of names and email addresses, "white pages". Although these services, such as Four11 [21] and InfoSpace [24], can provide coordinates by which to contact individuals, they do not specify particular interests or expertise. Various special interest lists provide a only a partial solution. On a large network, interest groups are numerous and rapidly changing. In response, we have implemented a system, ReferralWeb, which assists in locating people with specific interests or expertise. Our system also illustrates the network of social relations surrounding the user and expert. These relationships can not only be exploited to engender a response from an expert, but also serve as a reference for an expert's credibility.

First, we detail the motivation for developing this system. Next, we describe attributes of existing services and the context under which this system was conceived. Last, we delineate its objectives and compare it to those services.

1.1 Motivation

A natural method for locating resources is by querying one's informal network of personal relations: collaborators, colleagues, friends, acquaintances, etc. Initially, immediate relationships are exhausted. For example, a family member may recommend a trustworthy mechanic, or a colleague may offer advice beyond one's expertise. Often, if a direct relation fails to provide a sufficient resource, he or she will recommend another person who can. Thus, finding information or services is a matter of following personal referrals, a *referral chain*, generated from one's local social network.

Studies have shown that the social network is an effective channel for the dissemination of information and expertise [3]. The success of the social network in part is due to the "six degrees of separation" or small world phenomenon where any two individuals are separated by a small number of direct personal relationships. For example, within the electronic community, analysis of email logs by Schwartz and Wood [12] indicate that the average distance between any two persons is 5.4.

The limitation on publicly available information also contributes to the efficacy of personal networks. A person cannot record entirely the knowledge of his expertise, and often is reluctant to answer queries from strangers. A referral from a close colleague, however, provides incentive for a response. In this case, locating information can be accomplished only by traversing a referral chain from searcher to expert.

Regardless of its effectiveness, manually searching for an expert in a social network and engendering a response can be rather tedious and frustrating. Contacting too many individuals at each step for every query taxes relationships. For example, persistent emails to all of AT&T which don't involve most of the employees will eventually cause everyone to ignore the requests. Persistent individual emails to peers consume their time and strain their charity. On the other hand, as one narrows the set of contacts, the likelihood of failure increases. Further, experts are often professionals with little spare time to offer valuable advice. Thus, isolating an expert is relatively useless without a common colleague that can be exploited.

A common colleague with a desired expert serves a dual purpose. Most simply,

it provides an incentive for the expert to respond. As an extreme example consider Marvin Minsky, an expert in artificial intelligence. For him to consider my requests and respond within a reasonable time frame, I must be recommended by a trusted colleague of his. Oftentimes, a mutual colleague is also an implicit appraisal of an expert's credibility. For example, if one seeks an expert on programming languages, a colleague of his advisor is probably more credible than a friend's friend.

We have developed an interactive system, ReferralWeb, to assist and simplify the process of locating experts. It allows users to query for experts on keywords. It permits searches for experts within a specified radius from a given person, or locates a global expert by identifying frequent co-occurences with the given keywords in various document collections. By analyzing public data, this system not only identifies potential experts on user specified keywords, but also provides a list of relations by which to contact them.

1.2 Background

ReferralWeb falls into the generic class of systems which attempt to harness and manage the voluminous information available online. Most of these systems can be divided into two subclasses: search engines and recommender systems. We describe attributes unique to each category and illuminate the context in which our system was conceived.

Search engines encompass non-adaptive interactive systems which allow users to browse large, possibly dynamic databases. AltaVista [17], HOTBOT [22], and the Collection of Computer Science bibliographies [19] are examples of such systems. They utilize information retrieval techniques to isolate instances from a large collection that satisfy a query of some combination of terms. Various heuristics are applied to rank the fetched elements. Moreover, identical queries on constant databases return identical results. Their results are tailored to the average user rather than the individual. Meta-engines are systems which filter the output of generic search engines. Examples include services such as Ahoy! [18], which locates homepages, LawCrawler

[25], which searches law databases, and MetaCrawler [26], which combines the output of several web indexing engines. Although these services are more specialized than generic search engines, they still are tuned for the average user.

On the other hand, recommender systems aim to satisfy and adapt to the needs of the individual. The Boston Restaurant Guide and the Internet Movie Guide are some examples. These systems utilize collaborative filtering techniques, which analyze users' feedback of preferences, profiles, and/or ratings, to recommend restaurants, movies, music, etc. Items recommended are derived from ones preferred by users with similar interests to those of the given user. These recommendations evolve with users' preferences. FireFly [20] is unique in that it attempts to create new communities by uniting previously unrelated people with similar interests. Although the previous systems are limited to a focused set of topic areas, there exist adaptive services which recommend web pages and usenet articles. Fab, SiteSeer, GroupLens, and Phoaks are some examples [1], [11], [8], [14].

There are two significant drawbacks to recommender systems. First and foremost, most require significant initial investment on the user's part to be productive. Users must enter elaborate profiles or rank items in an iterated training process until recommendations become accurate and useful. Another problem is that no recommendations can be made if a given user's tastes do not overlap with others' in the community using the service. While these systems have been successful for recreational use, they are not economical for busy professionals.

Originally, in the spirit of recommender systems, Kautz et. al. [6] envisioned an agent based framework to assist and amplify person to person communication. It aims to satisfy keyword queries for experts in various fields by mimicking the referral chaining process. Each user employs a personal agent that scans word indexes of his or her email archives and a manually entered profile to determine expertise. The user profile is a list of contacts and keywords describing his and their expertise. If query keywords and terms in the profile do not produce a suitable match, the query is passed to agents of email correspondents or contacts. If these colleagues are not the desired experts, their agents in turn recommend additional contacts. The originating

agent iterates on suggested referrals until an expert is found or no recommendations remain. Essentially, this system searches along a referral chain until it finds a single valid expert. Such a search not only identifies an expert, but also suggests a path by which to contact him or her.

Simulation experiments and experience with a prototype indicated locating experts by keyword matches and referral chaining could be successful; however, some problems hindered the development of the prototype. Years of email logs were required for the system to be effective. People were unwilling to trust software agents with private email and a complete list of contacts. Last, a critical mass of participants was necessary in the agent's network for finding a reliable expert.

SixDegrees [27] is another effort that attempts to facilitate person to person communication. This project reconstructs the global social network by explicitly polling individuals for their occupation, location, hobby, skills, and social relations. Interests are specified from preset categories and relations are identified by name and email address. Users are solicited to participate via email. An elaborate constitution of rules are enforced to ensure user privacy and accuracy of reported relationships. Although this approach recovers an accurate network, privacy concerns limit participation and devolvement of relations. Further, the predefined areas of interests and skills only permit searches for people on broad, generic, and often recreational topics.

In the next section, we delineate ReferrralWeb's objectives and where it falls in the spectrum of these existing services.

1.3 System Description

We developed ReferralWeb, an interactive prototype which assists in locating experts on user specified topics by analyzing only publicly available resources. This system attempts to achieve two orthogonal goals. One aim is to locate people with interests or expertise on user specified keywords or topics. Unlike, Kautz's agent-based system, ReferralWeb's purpose is not just to find a suitable person, but rather provide a range of experts from which a user can choose according to his constraints. The

second purpose is to generate a useful referral chain by which a user can contact and engender a response from an expert. To achieve these goals, ReferralWeb automatically reconstructs and facilitates visualization and exploration of the underlying social network in existing communities. By instantiating the larger social network, a user can identify his place within it and discover connections to resources that would otherwise lay hidden over the horizon.

The intent of our system is not to replace generic search engines but complement them. ReferralWeb is a combination of a static search engine and meta-engine. It employs results from generic search engines to locate experts, and it utilizes a database of social relations for generating referral paths. Like recommender systems, ReferralWeb aims to satisfy the individual by generating a referral chain from expert to user. Unlike recommender systems, it is not adaptive and provides named referrals as opposed to induced, anonymous recommendations. Moreover, it requires no initial investment. Although this system suffers from the limitations of public information, eventually it could serve as a back-end for an agent based referral system. Such a hybrid could mollify privacy issues, overcome limitations arising from a dearth of email logs and participants, and augment network data derived from public sources.

At the implementation level, there are three distinct components to the current system: network constructors, search engine, and user interface. The network constructors are tools which reconstruct the global social network from document collections; currently, indexed web pages and one bibliography collection is examined. The search engine processes expertise queries by analyzing the social network databases and documents available on the Internet. The user interface is graphical front end which collects user queries and allows crude visualization of the reconstructed networks. Currently, the user interface is implemented as a Java Applet viewed in a web browser.

This thesis is a description of the framework, implementation, and effectiveness of ReferralWeb. Chapter 2 describes the assumptions adopted and general framework around which ReferralWeb is implemented. Chapter 3 discusses the data structures and algorithms used in the implementation. Chapter 4 delineates results of

experimental studies and key problems which challenge our assumptions. Chapter 5 concludes with drawbacks of our system, potential solutions, and suggests points of further development.

Chapter 2

Framework

An implementation of ReferralWeb's features requires precise definitions and axiomatic notions upon which to build. In the previous chapter, we presented intuitive notions for the terms: expert, social relation, and referral chain. We now present definitions that are motivated by practical aims. We enumerate and justify assumptions adopted to isolate experts and reconstruct the network of social relations from document collections. These assumptions simplify our problem, conforming it into a canonical information retrieval (IR) framework. Standard IR techniques can then be employed to initially implement this system (Chapter 3). Later, we will assess the validity of these assumptions (Chapter 4).

2.1 Definitions of Terms

ReferralWeb primarily assists users in locating persons with specific knowledge necessary for users' aims. Thus, we adopt the following functional definition for an expert.

> **Expert:** *A person, usually with specialized skills on a non-trivial topic, who can assist a user in achieving his end goals.*

Notice this definition is from the user's point of view. For example, to someone who knows nothing about programming, a mediocre hacker may qualify as an expert on C++. This perspective simplifies unnecessary complex factors involved in classifying

people as experts based on skill level, experience, etc. Our system is meant as tool to contact persons with potential solutions, not one to assess the proficiency of an individual. Although an expertise measure may be instructive, it is peripheral to this definition.

Social relationships are essential means for contacting experts. Ties such as friendship, peerhood, kinship, collaboration, and many others can be utilized to elicit a response from targeted individuals. Accordingly, we define a social relation.

> **Social relation:** *Any mutual, interpersonal connection between two individuals that may enable a third party to contact and elicit a response from either one of the two.*

This definition fuses the variegated connections that exist in a community and eliminates negative and directed relationships. Negative relationships such as "competitors" or "antagonists" are often useless in assisting users. Directed relationships, such as "knowing of" or "admires," are useful in locating experts; however, they provide little incentive for experts to respond. From this concept intuitively follows the definitions of *social network* and *referral chain*.

> **Social network:** *The set of social relations that exist within a community.*

> **Referral chain:** *Any path of social relations connecting any two individuals in a social network.*

These definitions provide a basis upon which we can discuss the framework and implementation of ReferralWeb. In the next section, we discuss a representation for a social network and methods for recovering it from a document collection. Last, we present methods for isolating experts.

2.2 Modeling and Recovering Social Networks

A social network is a collection of social relations that exist within community. An intuitive representation for a social network is a weighted graph, $S(V, E)$, where the

vertices, V, represent individuals and the edges, E, represent social relations. The weights of the edges indicate the strengths of relationships. A person's inclination to respond to a request is, in part, a function of his relationship to the requester. Also, the strength of a relationship often reflects the validity of a referral. Thus, the weights implicitly appraise the responsiveness and credibility of an expert. Since these strengths quantify an abstract notion of social distance between individuals, they only hold a relative significance. We now provide a classification framework for reconstructing the social network from a document collection, D.

For our purposes, persons are treated as objects and the documents that refer to them are features. Thus, associated with each person, p, is a feature vector, W_p, of length $\mid D \mid$. Each element W_{pi} of the feature vector contains a value corresponding to the appearance of the person in document D_i. Ideally, each entry indicates how strongly the document refers to the individual. We initially choose to utilize binary values, signifying whether a person was mentioned, for simplicity and reasons delineated later.

We make the following assumption to estimate the social distance between persons from their feature vectors.

> **Assumption 1** *The strength of a social relation varies monotonically with the number of documents within which both individuals appear.*

Consider bibliography collections in which the citations are represented as documents. Clearly, co-authorship is a relation whose strength increases with the number of common publications. Within unstructured collections such as indexed web pages, which contain homepages, faculty lists, organizational charts, etc., this assumption is still valid for an average person. Although this assumption is not completely accurate, we adopt it as a primitive notion. Thus, the dot product of a pair's feature vectors can serve as a crude estimate of social similarity. On binary vectors, this computation only contributes a positive value for features a pair have in common, thus counting the number of documents in which both persons appear. Let us consider how to recover the social network using this metric.

If each person's feature vector is consolidated into a person-document matrix, $\mathbf{W}$, with rows corresponding to unique individuals and the columns corresponding to documents, then the square of that matrix, $\mathbf{S} = \mathbf{W}^2 = \mathbf{W}\,\mathbf{W}^T$, yields a person-person matrix representing the social network graph, $S(V, E)$. An element $\mathbf{S}_{ij}$ contains the number of documents persons i and j have in common, their social similarity.

This distance measure, however, biases the tie strengths toward people who appear in the document collection more often. Consider again Marvin Minksy, whose expertise spans numerous fields. He may appear in 2000 documents total with about 50 documents in common with many distinct persons. A pure product would rank any such pair with Minsky higher that two people who have 25 documents total, with all 25 documents refering to both. To solve our "Minsky" problem, it is essential to use similarity measures normalized by the magnitude of feature vectors. The Jaccard, Cosine, and Dice coefficients are simple metrics which have this property [2]. In our implementation, we utilize the Jaccard coefficient, the ratio of common features to size of their union, for purposes of convenience. This corresponds to dividing each element $\mathbf{S}_{ij}$ by $(|W_i| + |W_j| - \mathbf{S}_{ij})$. In the context of clustering, these measures yield equivalent results, even with the use of weighting schemes for feature vectors [2]. Further, simple similarity measures are often monotonic with more complex ones [2]. Since we are only interested in relative comparisons, we cannot justify more intricate metrics.

To recover $\mathbf{W}$ and compute $\mathbf{S}$, names of persons need to be identified within documents. This task is trivial in a structured collection such as bibliography citations which label authors, editors, etc. However, for unstructured text, a variety of techniques developed in the message understanding community for extracting names can be utilized [13]. Finally, most positive entries in $\mathbf{S}$ represent edges of real social relationships; weak edges unlikely to exist are pruned by applying a threshold to the final weights.

2.3 Finding Experts

Given a query on some keyword(s) or topic, our system attempts to find a person with the appropriate expertise. Within this section, we will use the terms keyword and topic interchangeably. ReferralWeb permits a variety of such keyword queries. The following assumption is central to the implementation of each variation.

> **Assumption 2** *The expertise level of a person on a given topic varies monotonically with the number of documents in which both the keywords and person appear.*

We justify this assumption by considering some examples. In a bibliographic collection, the more an author publishes on a given topic, the stronger his expertise in that area. Our assumption holds for web pages containing names since they are often homepages or listings of people with common interests. However, this assumption is weak for usenet postings and articles. Often, persons with expertise in an area are too busy to post, while mediocre hackers offer invalid advice. Regardless, we adopt this assumption as a primitive and avoid mining information repositories for which it is invalid.

We utilize a standard framework to rank persons as experts on given topics. As before, we represent people with document indexed feature vectors, W_p. We view topics as objects and documents that contain them as their features. Similarly, we represent topics with binary vectors, K_t. Thus, given a list of both topics and people, by consolidating the vectors into matrices, we derive the person-topic space, $\mathbf{Z} = \mathbf{W}\,\mathbf{K}^T$. The rows rank the topics as areas of interest or expertise for each person. The columns rank the individuals as experts on a given topic. Again, an element-wise operation is necessary to normalize the weights. In a typical query a single or at most a few keywords are given.

Isolating an expert involves ranking each individual's feature vector against a given topic feature vector using some similarity measure. Since ranking is more crucial in this case, a variety of normalized metrics may be optimal [2]. In our implementation, we again utilize the Jaccard coefficient for simplicity and pragmatic considerations

specified in the next section. In the previous section, methods for recovering W_p are given. Standard IR techniques are used to recover K_t.

We have described the skeleton of methods and representations upon which ReferralWeb is built. Note, as apposed to the standard IR model, this model treats names and keywords as objects while documents are their features. The described framework is abstracted from the issues involved with data structures, algorithms, and information sources used in the implementation. In the next chapter, we detail our system and illuminate crucial problems that arise.

Chapter 3

Implementation

ReferralWeb consists of several components which implement the framework outlined previously. The fundamental computation in both reconstructing social networks and isolating experts is the identification of names within unstructured documents. Initially, we describe the heuristics used for this task by the name extractor subcomponent. The underlying social network can be discovered from document collections using this name extractor. We delineate the procedure utilized by the network constructors to recover social networks from the DB&LP bibliography collection [23] and indexed web pages. In this implementation, these are the only two copora we mine. For development purposes, we limit ourselves to persons associated with the computer science community. The search engine component also utilizes the name extractor and the social networks for three types of queries. We motivate and outline how the search engine executes these queries. Then, we describe the user interface and demonstrate how it facilitates visualization of the underlying social network. Finally, we specify the software design of this system and some of its drawbacks.

3.1 Extracting Names

Extracting names from the unstructured documents on the web is a difficult task. Techniques developed in the message understanding community extract names with high precision and recall (better than 90%); however, most require large natural

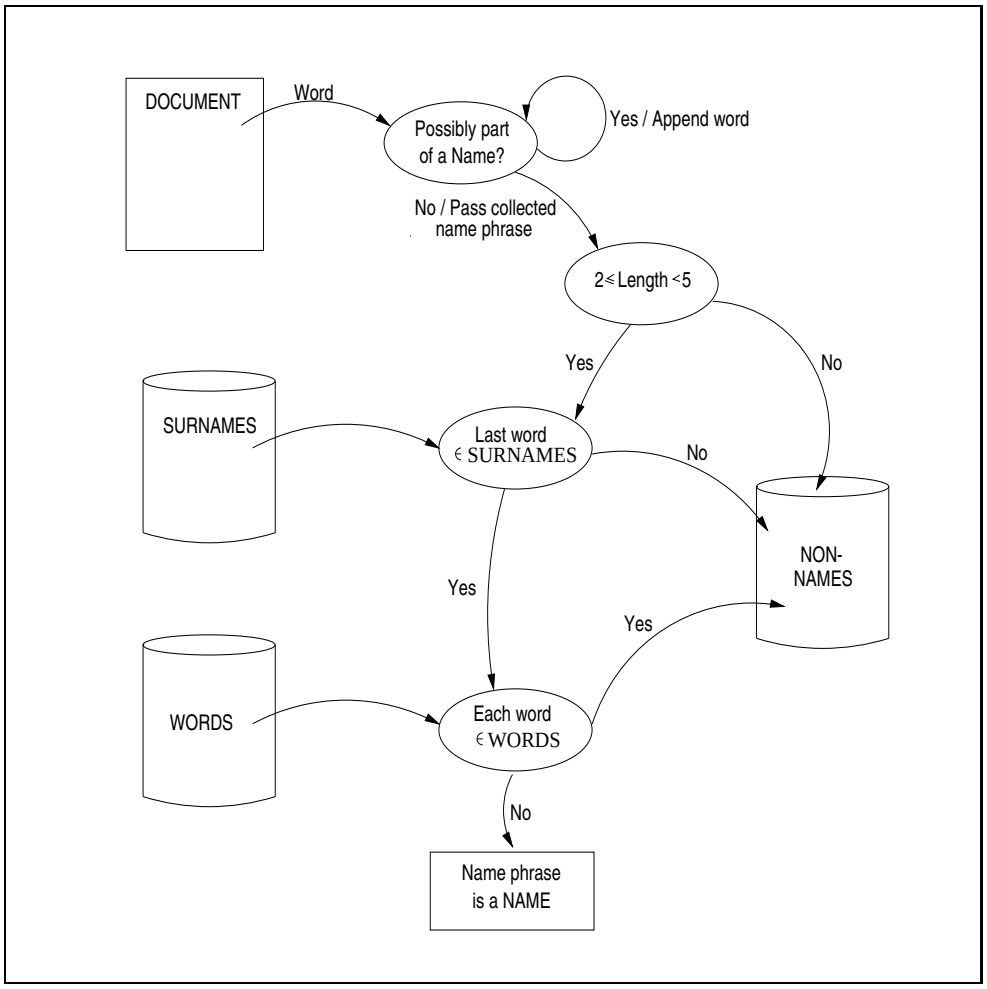

Figure 3-1: This diagram depicts the heuristics utilized by the name extractor.

language processing systems which are difficult to reproduce, acquire, or re-train [13]. In addition, these systems are tuned to extract names from semantically structured English corpora such as articles from the Wall Street Journal, not typical HTML documents. Thus, ReferralWeb implements a simple pipeline of heuristics as shown in figure 3-1 inspired by the Basic NE system developed at CRL/NMSU [13].

First, all HTML tags are filtered out of a document. Then, sequential strings of capitalized words are extracted from the document into a phrase. Separators are any non-capitalized tokens other than hyphens, apostrophes, periods, and inter-name words like “de”, “von”, etc. Phrases longer than five words are discarded because they are unlikely to be a name. The last word of the phrase is then filtered through

a dictionary of surnames compiled from public domain sources. Then all words are checked against a dictionary of words that are not only names. If they all exist in the dictionary, then the phrase is unlikely to be a name. Each phrase that survives the previous decision tree is assumed to be a name. Observations indicate that the capitalization and surname dictionary heuristics are the most significant steps in identifying names.

The feature vector associated with each person is indexed by documents. For each document processed, the name extractor subcomponent computes a single element in this feature vector. Therefore, it serves as the underlying routine utilized for both the network constructors and expert search engine.

3.2 Extracting Social Networks

The largest and richest accessible document collection is the World Wide Web. However, only a fraction of this collection actually contains data pertinent to existing social relations. Because the network is the main bottleneck in recovering these relationships, we must employ an algorithm which locates, retrieves, and analyzes this relevant fraction and avoids the rest. Thus, we implement a breadth first procedure which incrementally grows the social network from a known individual by mining documents containing that person and moving on to documents containing his social relations. This technique not only permits analysis of the most relevant documents, but also eases the evaluation of extracted relationships by begining from a familiar community. To locate documents relevant to a specific person, we probe a generic search engine because it has indexed most web documents, and it is infeasible to recreate it. Currently, we utilize AltaVista [17] because it appears to be the fastest and most comprehensive index of the web. First, we specify the procedures for extracting the social network from the web and bibliography collections. An outline of the representation used for the social network graph follows.

An initial global social network is constructed by repeatedly probing a generic search engine as depicted in figure 3-2. First, a known individual who has a significant

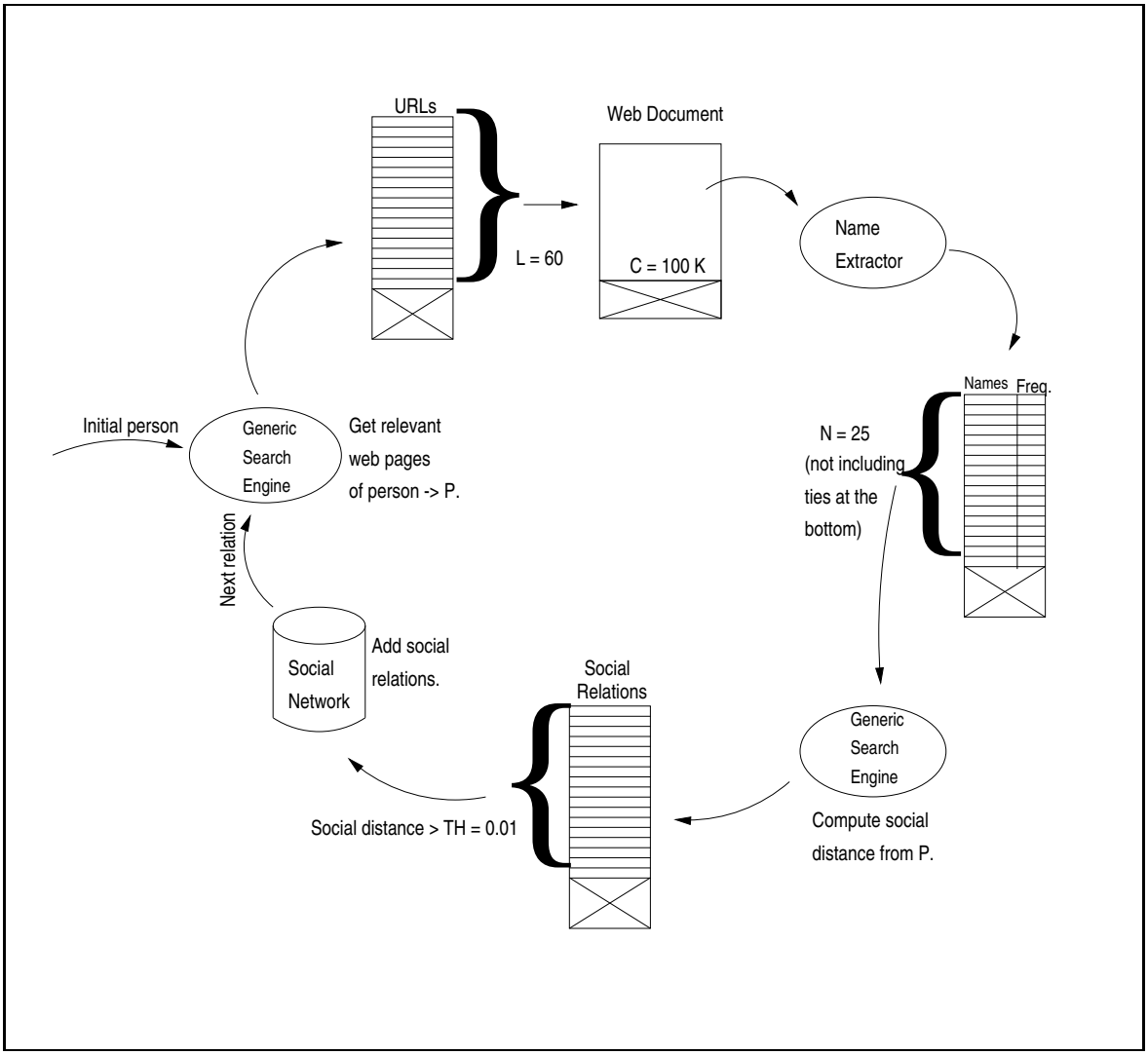

Figure 3-2: This diagram depicts the data flow in the breadth-first web-based network constructor.

presence on the web is selected. Then, the generic search engine is queried with his name as keywords. The number of documents referring to him is stored, and the top L ranked documents are retrieved from the locations specified by the generic search engine. Names and frequencies of their occurrence in these L documents are extracted. The most frequently occurring individuals up to a specified maximum N are retained as potential social relations; ties at the bottom are also retained.

Note, only the top L documents containing a name are retrieved and analyzed. Hence, the validity of these potential relations are highly dependent on the rankings of the underlying search engine as well as L. The top ranked documents are enough to identify close colleagues, but not complete enough to estimate overall social distance. Hence, a second pass through the potential relations probing the generic search engine is required for computing distances.

The social distance between a pair is computed via boolean queries; a feature provided by most generic search engines. The number of documents in which both individuals appear serves as the dot product of their feature vectors. A boolean "AND" query to the generic search engine with the names of these two as keywords estimates this value. The total number of documents referring to either individual can be computed via multiple keyword queries with each name specified separately. The ratio of these values is the Jaccard coefficient for the pair of individuals.

Using this similarity metric, we estimate the social distance between the given individial and each of the potential realtions and reject people under a threshold TH. The remaining persons are added as social relations. The number of common documents referring to both the given person and each relation is also stored. This entire process is applied recursively on the relations in a breadth first manner until no new individuals are found or the routine is manually terminated. Large documents with few names are a bottleneck because they contribute little additional information for the time wasted in processing them, thus we use a maximum cutoff C for the document size. The parameter values used in practice are $L = 60$, $N = 25$, $C = 100\mathtt{K}$, $TH = 0.01$ (for the Jaccard measure). The threshold TH and N were determined through surveys explained in the next chapter. The other parameters were adjusted

in a hunt and peck fashion until valid relations for known individuals were recovered. These values will most likely vary depending on the generic search engine and community being reconstructed.

The reliance on the search engine elucidates the considerations for assuming binary feature vectors and using a simple distance metric. Search engines often do not offer weights for the rankings of their documents on given keywords. The only information rankings offer are ordinal relations. Within these rankings there may be several equivalence classes which also are unspecified. Some search engines, such as HOTBOT [22], provide normalized weights for returned documents. However, the significance of these weights is unclear. Even with our pessimistic assumptions, we are able to recover relevant social relations from the web.

Another rich source of social relationships is bibliographies. Although these databases are restricted in the breadth of people and topics they cover, they contain stronger direct evidence of relationships (collaboration). Currently, we have processed the Databases & Logic Programming bibliography [23]. For this collection, we scan the database sequentially augmenting the social network as relationships are identified. In this case, each citation represents a document. Since the citations are short and only provide a dichotomous indication of relationship, feature vectors with only binary elements suffice. The total number of citations for each person, and the number of common citations for each relation are stored. Again, these values can be used to determine the strengths of ties. No pruning of relationships is performed since collaboration is usually a valid social relation. The social distances now emphasize "closeness" rather than validity of relationships.

A convenient representation for the global social network, $S(V, E)$ is a symmetric adjacency list. Even though there are extreme cases like Marvin Minsky in which some individuals are highly connected, on average, a person only has about twenty to thirty relevant *social relations*. Because S is sparse, memory considerations make this representation ideal. Within each vertex or node, V, representing a person, we record his or her last name, and all first and middle names. The name uniquely identifies a single individual. Each node stores the number of documents which refer

to the person. Moreover, associated with each node is an adjaceny list of nodes and weights containing each social relation and the number of documents within which both appear.

3.3 Performing Queries

Three tractable classes of queries for locating experts are implemented by ReferralWeb: path, global, and local. The first type simply finds all shortest paths or referral chains between two specified persons. This query is sufficient if one knows from whom to obtain expertise. If the desired expert is unknown, one can resort to the global query. The global query locates experts by mining indexed web pages which contain user given keywords. Since this mechanism is susceptible to finding unreachable experts, one can resort to the last option. The local query attempts to identify the best expert on keywords within a specified social radius of a given person.

The simplest query is for users that already know of an expert and require a path or several paths by which to engender a response. Initial surveys delineated in the next chapter indicate these weights have little relative significance. Thus, although social distances are recorded within the reconstructed networks, the current implementation of this search ignores these weights. A breadth-first algorithm is employed for computing shortest paths. This query mechanism is often used in conjunction with the expert queries for obtaining a valid referral chain.

In the case a user has no knowledge of an expert, a global keyword query mechanism is available. For this option, the user must specify keywords, the number of documents to examine, L, and the number of experts to return, R. The latter two parameters are necessary to limit the running time of the query. This system attempts to locate experts by analyzing relevant web pages in the following manner. First, the system probes the generic search engine with the given keywords. Similar to the network constructors, the top L ranked documents returned by the generic search engine are retrieved. Using the name extractor, all extracted names are sorted by the frequency of occurance within these L pages. Since only the top ranked documents

are mined, the top R names serve only as potential experts.

Another pass through these potential experts probing the generic search engine is required for estimating their expertise level. Using the same boolean query function employed for estimating social distance, the dot product of the feature vectors for potential experts and keywords is computed. In this case, the conjuncts in the query are the keywords and the names of potential experts. These names are then ranked by their Jaccard coefficients with respect to the keywords and returned as experts. As mentioned before, in our framework, these coefficients estimate the expertise level of each individual. This procedure attempts to identify the best experts from within the document collection. However, there is a chance they do not exist within the reconstructed social networks or they are unreachable.

In such a situation, a user must resort to the local expert query. This mechanism attempts to rank all persons surrounding a specified individual within a given social radius. Since social strengths are ignored, the social radius merely specifies the maximum number of steps from the given person within which to consider neighbors as experts. In a breadth-first fashion, the dot product of the feature vector for each neighbor with the feature vector for the keywords is computed via the same boolean query function for estimating social distance. Again, the conjuncts are the keywords and the names of the neighbors in consideration. Once all the neighbors within the social radius are considered, they are ranked by their Jaccard coefficients with respect to the keywords and returned as experts.

For both types of expert queries, rankings can be improved if a document scoring scheme for computing social distances and expertise level is utilized [2]. The difficulty of re-indexing the web and the lack of functionality in generic search engines prevents us from using a more complex metric.

3.4 Visualizing a Social Network

One of our subgoals is to allow exploration of existing relationships within a community's underlying social network. This is accomplished by allowing the user to

traverse a local social network and extend its boundaries. We have developed a unique intuitive interface, implemented as a Java Applet, for displaying and interacting with a 2-D graphical representation of a social network.

To visualize a local social network, the user interface initially queries the user for a starting "anchor" person from which to explore the network. Then the immediate network of that person is displayed in a sub-window of a web browser. Each person is represented by a rectangle containing his surname. Social relations are represented by lines connecting these rectangles. The original placement of nodes is random, and the "anchor" node is fixed. Then for each iteration of the event loop, the interface attempts to find a better placement of the nodes by utilizing a spring model for the connected nodes. To prevent nodes from overlapping we incorporate an additional repulsive force when their encompassing rectangles overlap.

Th dynamics of such a model can be approximated by minimizing the following potential with respect to the positions of the non-fixed nodes.

$$\Phi = \sum_{(u,v)\in E} \left(\delta(u,v) - \delta_0\right)^2 + \sum_{u,v\in V} \square\left(\delta_r(u,v) - \delta(u,v)\right)\left(\delta_r(u,v) - \delta(u,v)\right), \qquad (3.1)$$

where E and V are the sets of edges and nodes displayed; δ_0 is the ideal distance between connected nodes; $\delta(u,v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2}$ is the distance between nodes u and v; $\delta_r(u,v)$ is one half the sum of the diagonals of the rectangles encompassing u and v; and $\square(x)$ is the unit step function.

This potential contributes a quadratic term for connected nodes that are displaced from their ideal distance, similar to a physical model for springs. It also contributes a linear term for nodes that overlap on the screen to unclutter them. A hill climbing computation is performed to minimize the above potential. First, the derivative of this potential with respect to each node's horizontal and vertical position is computed at each time step. This vector is then normalized and scaled by a temperature factor. Then the nodes' positions are incremented with the corresponding values from the scaled derivate vectors.

The derivative calculation requires E steps for the contribution from the connected nodes. In addition, two lists of the nodes are maintained, one sorted by horizontal position and another by the vertical position. These lists are scanned sequentially for pairs that might be within ten percent of the total distance between the end nodes. The contribution to the derivate from the overlap term is then calculated for those nodes. Although, this is not an accurate derivative calculation, it suffices for our purposes. Thus, the total derivative operation takes $O(E + V)$ steps.

The user interface allows the user to interact with the graph as it is being "minimized". This permits the desired aesthetic placement and untangling of the graph when the hill climbing routine falls into a local minima. To explore the boundaries of this network, the user can extend the graph by selecting the appropriate option and node. The hill climbing continues as additional nodes and edges are added. The newly extended node is "anchored". Repeating this procedure, the user can walk the social network around a given person. A path query between two individuals returns the nodes and edges along a path which is displayed with the endpoints anchored. The user can also walk the social network along this path.

One final feature worth noting is "details". This option allows the user to look at the details of the person via additional browser frames linked to Ahoy! [18], which locates the person's home page, and AltaVista [17], which lists web documents relevant to the individual.

3.5 Software Design

ReferralWeb, whose original aim was to provide a "proof of concept", is written entirely in Java 1.0.2 and consists of several modules. The network constructors are a group of programs operated independently of the rest of the system. These modules require the most disk space and network bandwidth to recreate social networks at a reasonable speed. The search engine is a server which consists of two modules, one which serves information about persons and their local social networks and another which processes queries for experts. These modules require the most memory because

they hold the entire social network databases in memory for speed. They also utilize a significant portion of network bandwidth to process multiple users' queries. The user interface is a Java Applet which is served by the same machine upon which the search engine operates. This allows the user interface, typically running within a web browser, to connect to the search engine's server and retrieve results for user queries. The interface is meant to be lightweight and transportable, thus requiring minimal bandwidth for retrieving query results and some memory for display the network.

One major disadvantage of ReferralWeb is that the network constructors and search engine are written in Java and compiled to byte-code. The byte-code is interpreted which inherently adds space and time overhead. Furthermore, objects and structures in Java are more costly than ones in C. For each network constructor to function efficiently, each alone requires the significant portion of a 128 megabyte SUN UltraSPARC on a three megabit/sec network connection. The search engine servers also need to execute alone when serving a social network database of 10,000 nodes or more because they consume the system's memory resources. The network constructors and search engine servers should be rewritten into a compiled language. A user interface implemented in Java is ideal because it allows portability of the code and accessibility of the system to numerous platforms without recompilation.

Chapter 4

Evaluation and Discussion

The effectiveness of ReferralWeb was primarily measured through personal interviews, user feedback, and practical experience. A survey combined with personal interviews helped ascertain the validity of the social relationships extracted from indexed web pages. Observations from these interviews also shed light on the utility of the bibliography collection as a source of social relations. A simple experiment in searching for experts on various topics indicates the localized expert query is much more effective than the global. Furthermore, no other service or combination of services can fully substitute for ReferralWeb's function of easing the identification of experts. Unfortunately, only anecdotal evidence supports these evaluations.

In the context of our experiments, this evaluation isolates a few critical drawbacks. The network constructors and search engines perform no document scoring of their own, making them heavily dependent upon the generic search engine's rankings. The name conflation issue is essential in recovering real social relationships. Aspects of relationships such as the age and nature of ties which are indicators of social distance are not modeled. Finally, strength measures would be useful in pinpointing experts. In the next chapter, we discuss some potential approaches to remedy these shortcomings.

Measurement	DK	DB&LP
No. Persons	1161	37881
No. Explored Nodes	191	37881
Avg. No. Relations	17.17	3.612
Var. No. Relations	35.81	28.07

Table 4.1: This is a comparison of the networks extracted from the DB&LP bibliography and indexed web pages starting from David R. Karger.

4.1 Social Network Databases

Table 4.1 summarizes the nature of the social networks extracted from the web centered around David R. Karger (DK) and the Database and Logic Programming bibliography collection [23] (DB&LP). Explored nodes are the nodes for which social relations have been extracted by the breadth-first web-based network constructor. The average number of social relations per node and the variance in the number of social relations are computed only for the explored nodes since unexplored nodes clearly will lack many relevant social relations. From these figures, it is evident that the DB&LP network is much sparser than the DK network. The high variance in number of neighbors from the DB&LP network indicates that it may not reflect many relationships which actually exist between pairs within the database.

A simple computation was performed comparing the nodes and edges existing within both databases. The DB&LP network contained 738 nodes present in the DK network. Among those nodes, 3611 edges or social relationships existed in the DK network. Of those relationships, only 1796 existed in the DB&LP network, a little under 50%. Although, these additional relationships could be erroneous, user surveys indicate that the precision of the web-based constructor is better than .50. Thus, it appears the DB&LP network is a weaker model of existing relationships. The advantage to the bibliography based network, for this specific implementation, is that it contains many more persons than the web based network.

4.2 Validity of Social Networks

To ascertain the validity of extracted relationships we performed two phases of surveys. The first phase involved a survey intermingled with an informal personal interview. This phase helped set the parameters for the web based network constructor. It also provided fundamental insights into the problem of mapping relations. The second phase verified and appraised the relationships extracted from both document collections after the tuning. From these experiments we aimed to estimate the precision, recall, and accuracy of relationship rankings for the web based network constructor.

There are two notions of precision and recall for extracted relationships we could measure: global and local. The global precision is a measure of the relationships recovered throughout the entire database, the ratio of the actual relationships recovered to the total number of relationships recovered. The local precision is the ratio of the actual relationships recovered of some single individual to the total number of relationships recovered for him. The global recall is the ratio of the actual relationships recovered to the total number of actual relationships that exist between pairs in the database. The local recall is the ratio of the actual relationships recovered for some person to the total number of actual relationships he has. We chose to estimate the local measures which aided in setting the parameters for the breadth-first web-based network constructor.

In the first phase, each individual interviewed was asked to identify social relations from a list produced by the web based network constructor. Fifty extracted names were listed in no particular order for each individual interviewed. Each interviewee was asked to perform three tasks with this list. First, he was asked to mark each name that was a relevant social relation. A social relation was defined as someone to who will consider a recommendation made by the interviewee for some third party or vice-versa. These identifications determine the precision of the network constructor. Next, he was asked to group the marked individuals into equivalence classes and rank these from strongest to weakest relations. Using the *ndpm* measure from [16], these rankings estimate the accuracy of the distance metric. Finally, the interviewee was

	Before	After thresholding		
Person	Precision	Precision	Recall	*ndpm*
Leiserson	.64	.88	.88	0.30
Karger	.63	.74	.96	0.33
Selman	.43	.81	.95	0.40
Frigo	.16	.20	1.0	0.45

Table 4.2: These are the results of the first phase of surveys.

Person	Precision
Rivest	.90
Viola	.65
Galperin	.75
Maron	.30
Kautz	.12
Isbell	.00

Table 4.3: These are the results of the second phase of surveys.

asked to list any relevant relations that were not listed from which we could estimate the system's recall.

We interviewed four persons in the computer science community ranging from high profile professors to a graduate student. The effectiveness of the network constructor varied greatly. As expected, for high profile cases, the precision was almost perfect. For the graduate student, only a few names were even recognizable. In addition, the rankings offered by the constructor had little significance, performing just a little better than a random permutation ($ndpm = 0.5$). From interviews, it appeared that the rankings made by individuals were dependent upon intangibles such as personality, length of relationship, etc. The interviewees also found the definition of social relation contrived and unnatural. Some declared they would use their own intuitive notion of social relation. Lastly, it was impossible to measure recall. Interviewees were both unwilling and unable to list or count the numerous relationships that could potentially be used for references.

From these interviews, we were able to estimate a threshold distance at which we

could reject candidates as social relations. We set $TH = 0.1$ for pruning relations by maximizing the sum of precision and recall from only the names listed and marked by each interviewee (see table 4.2). For the next phase we ignored both the rankings and recall. We set the number of neighbors returned from the name extractor to $R = 25$, the average number of individuals positively identified as relations among the first four. We extracted names for six other individuals and obtained consistent results. Graduate students had low precision, established professionals had much better results (see table 4.3). For one graduate student, almost no real relationships were identified. This is because his identity on the web is as a hip-hop music critic. Commonly appearing names with his included only famous artists.

These surveys helped identify several sources of errors for the network constructor which we list below.

- First, the name extractor often misidentifies common proper names such as Addison Wesley, New York, and San Diego as names. Another database filter containing these commonly mistaken phrases may help. A parser for marking dates would also eliminate many irrelevant names. Eliminating irrelevant names increases the probability of relevant names surfacing to the top of the list of possible relations.

- Name conflation is also a huge factor. Social distances for persons are often distributed among variants of the same name causing the name constructor to ignore strong relations.

- Since the constructor performs no document scoring of its own, it is highly dependent upon rankings provided by the underlying generic search engine. Thus, for people with little presence in web collection, this system is susceptible to pages with long lists of names. An inter-document term weighting scheme might improve but definitely not worsen the quality of extracted relationships.

- The social distance metric is vulnerable to duplicate documents on the web. This was the case with Kautz, where his name appears on a list of contributors

to GNU Emacs. This list is duplicated several times, thus giving irrelevant names higher scores.

- For some individuals, their identity on the web is completely different from their identity in real life. Thus, their estimated relations can only reflect their identity on the internet as was the case for the hip-hop critic.

- Lastly, there are numerous existing relationships that are missed because the document collection contains no evidence of them.

We compared the relationships identified and ignored from our initial interviews with the relationships in the bibliography database. Contrary to our initial expectations, the social network from the bibliography collection is a weaker model of relationships than the network reconstructed from the web collection. Although the relationships are often accurate, the bibliography collection is simply not complete. As is evident by the comparisons, the bibliography collection misses about 50% of the relationships extracted from the web, which is worse than the local precision of the web-based constructor for established professionals. The local precision is a good estimator of the global precision. Thus, it appears bibliography network is sparse and misses many actually existing relationships between pairs in the database. Furthermore, prolific authors tend to have numerous weak relationships. In contrast, for this specific implementation, the bibliography collection contains many more individuals. Thus, it is likely a relevant expert is already in the network database allowing the system to recover some path to him. The accuracy of the web based network is partly because the range of documents mined encompasses some of these bibliography citations.

4.3 Effectiveness of Search Engine

To measure the retrieval effectiveness of the expert queries, a simple experiment was performed. We constructed twelve keyword queries for various areas of computer science. Of those twelve, six were chosen randomly to be executed via ReferralWeb and

the other six were independently performed. Each potential expert was identified as an expert through his homepage or through relevant documents on the web. We provide anecdotal evidence for the effectiveness of our system. Furthermore, we describe the differences between the global expert search and local expert search mechanisms. As mentioned previously, the global search mechanism attempts to identify experts by simply mining, in the same manner as the network constructor, the top ranked web pages returned from the generic search engine on some given keywords and returning the top matching names. The local search mechanism identifies experts by performing a breadth first search around a given individual, evaluating each neighbor's relevance to the given keywords.

With ReferralWeb, for each of the six queries, we were able to identify potential valid experts through common colleagues of ours. Initially, a person that we knew existed in the database was chose as the starting point. These people included advisors, professors, as well as colleagues. Next, a localized search query within a radius of two (second-order zone) was executed. The top ranked persons returned from this query were searched for in Ahoy!. If suitable homepages were found, they were examined. Otherwise, AltaVista was consulted with their names and the top entries examined. Identification of an expert was rather subjective. Homepages, biographical sketches, and published papers and books served as evidence of expertise. We often verified whether the returned experts were professors or individuals from respected research institutions. If the local query failed to return relevant candidates, a global query was attempted. Again, the returned candidates were examined through Ahoy! and AltaVista. The experts in each category are recorded in table 4.4, including the method by which they were located, and the means for contacting him or her.

We made a few observations about the variations between local and global expert queries and their common vulnerabilities. Local queries are susceptible to the "Marvin Minsky" phenomenon. Often, renowned computer scientists came up as experts on various queries. People like Minsky are referenced across so many topics, they permeate as experts on many different queries. Although they could potentially be experts in those fields, we disqualified them because they are often difficult to reach

Topic	Expert(s)	Database
databases	Hector Garcia-Molina	DK
randomized algorithms	Rajeev Motwani	DK
	Prabhakar Raghavan	DK
probabilistic reasoning	Judea Pearl	DK
	Stuart Russell	DK
complexity theory	Avi Wigderson	DK
	Juris Hartmanis	ignored
compiler theory	Ravi Sethi	DK, DB&LP
simulated annealing code	Lester Ingber	none

Topic	Query	Path
databases	local, r=2	Karger-Koller-Molina
randomized algorithms	local, r=1	Karger-Motwani
	local, r=1	Karger-Raghavan
probabilistic reasoning	local, r=2	Karger-Koller-Pearl
	local, r=1	Selman-Russell
complexity theory	local, r=2	Karger-Nisan-Wigderson
compiler theory	global, path, local	Karger-...-Sethi
simulated annealing code	global	unknown

Table 4.4: These are results of expert searches with ReferralWeb.

even through common colleagues. Furthermore, their status allows them to be well connected. Hence, there was often a path through them by which one could contact a reachable expert. These paths were eliminated as well. As possible fix is to weight relationships normalizing by the number of relations a person has.

The localized query was quick and rather useful in identifying close experts. However, if no experts exisited within the second-order zone, the local query was useless because the system had to cycle through hundreds of names. At the time, we found ourselves performing a localized best-first search for an expert, simulating the referral chaining of Kautz's agents. Starting from some initial person, we would identify a neighbor with similar interests in the desired field and perform local searches around them. This pruning can potentially be automated and included in the next version of ReferralWeb.

Because this system is a prototype developed for correctness rather than performance, global queries took between 20-30 minutes to examine 40 documents. However, they were necessary in certain cases when localized queries could not produce a suitable expert. For the simulated annealing question, the best person found did not exist in either database. The global query produced a relevant expert, but our system offered no path by which to contact him. The global query was also employed for the "compilers" query.

As with most retrieval systems, the nature of the query also plays a role in the effectiveness of the search. Queries such as "compilers" were too broad and identified too many people as experts. For example, people working on new programming languages as well as people working on parallel compilers were identified as experts. We narrowed the topic to "compiler theory". Once an expert was identified through the global query mechanism, he was found in the bibliography based network. By exploring a path around that expert, a closer (to a common colleague), and more suitable expert was identified using the local query. In this case, we found Ravi Sethi, the author of an introductory book on compilers. There were several paths by which to reach him. Some paths could be eliminated because they involved traversing a renowned expert. However, three paths remained and it was unclear which one was

the best. A relevant responsiveness measure would have been useful.

The effectiveness of the name extractor heavily impacted the performance of global queries. Besides the common errors in extracting false names, global queries are biased toward signatures. A query for "compilers" resulted in identification of "webmasters" as experts because they owned pages with information on various compilers. One potential solution is to give higher weight to keywords that are closer to the extracted name. Global queries also were susceptible to errors where long lists of names appeared in conjunction with various topics. A person working in a field related to a specific topic often surfaced as an expert. Also, global queries were susceptible to the "Marvin Minsky" problem. Again, some form of document weighting scheme would eliminate most of these problems.

Identifying experts through using standard web and bibliography search engines was simple. Many valid experts on various topics exist on the web; however, to find means of locating a valid expert, we had to constrain ourselves to organizations to which we belonged. Manually extracting a referral chain using these existing services was much too tedious.

One unbiased graduate student also volunteered to use our system. His specific query was in the area of information retrieval. Even with the limited size of our database, he was not only able to locate an expert at a different institution, but also received an adequate response from the expert through the referral path he found. He mentioned an important aspect which our system neglects. Although he was able to obtain an answer through the expert identified by ReferralWeb, he could have just questioned a co-worker in the same building. Our system did not identify the co-worker because he did not exist in the databases. More importantly, even if the co-worker existed in our system, ReferralWeb does not account for physical or organizational proximity which is often the initial characteristic exploited to contact an expert.

In summary, the local query mechanism is the most useful in identifying potential experts within the second-order zone. The global query mechanism by itself is rather weak due to its slow speed, errors in identifying names, and susceptibility to single

references in lenghty documents. However, it can be combined with the local query mechanism to isolate a more specific expert. Moreover, our system is potentially better than existing services since it provides paths to unknown experts. The validity of generated referral chains is still unclear.

Chapter 5

Conclusion

We have described and developed ReferralWeb, a system for identifying and contacting experts on keyword queries. As a prototype, this system exhibits the potential for achieving its initial goals. The network constructors identify relevant relationships. The local search mechanism is indispensable for locating experts. The global mechanism, though more tedious and less robust, serves as a backup. However, as delineated in the previous chapter, the system has several key shortcomings that need to be addressed. The name extractor needs improvement for excluding dates and certain proper names. The issue of name conflation is central to identifying distinct individuals. The social distance metric does not adequately reflect the strengths of ties which are important in finding a useful referral chain. Local queries beyond the second order zone and global queries are too slow to allow constructive interaction. Finally, our system has no notion of physical or organizational proximity. In this chapter, we discuss possible solutions to these drawbacks and offer points of further research and development.

5.1 Identifying Individuals

5.1.1 Extracting Names

The name extractor is limited in that it does not account for geographical names, organization names, commonly capitalized non-names and dates. The obvious solution is to manually create a list of these phrases and use them as a negative filter in the pipeline of stages of the name extractor. However, this is not the final step. People are often referred in pages via pronouns. Some names like Cotton Seed will automatically fall through the name extractor since both components are commonly non-capitalized dictionary words. A more sophisticated name extractor with natural language processing capabilities is necessary. Such an extractor might be able to tag pronoun references, eliminate non-names, and identify names not present in the dictionaries based on context. Using a probabilistic model for names occurring within certain grammatical structure is another possible approach.

5.1.2 Name Conflation

There are two essential issues involved with extracting names from documents to identify individuals. First, individuals are often referred to by various names and pronouns. For example, David Karger may also be referred to as David R. Karger or David Ron Karger. Or more severely, Richard Karp is also known as Dick Karp and Frank Thompson Leighton also goes by Tom Leighton. In constructing social networks, these variants must be identified, otherwise weights for relationships will be divided among these variants allowing weaker relations to surface. The same holds for finding global experts on a given keyword. This problem occurs in web pages as well as bibliography collections, since authors often publish under various contractions of their names.

To identify and conflate names referring to the same individual we can assume variants of a name with at least one common social relation or a common expertise refer to the same individual. It is unlikely that a person knows two people referred

to by variants that are valid social relations. Thus, when extracting relations from web pages, for each individual, the variants in the list of potential relations can be conflated. For bibliography collections, names can be sorted by surname and variants identified. If the intersection of the neighbors of a variant pair is non-empty, their vertices can be contracted.

Although unlikely, under this assumption, there is a chance that a transitive conflict may arise. For example, M. A. Shah, M. Shah, and M. B. Shah may all be names with evidence of interests or expertise in the same field. It is unclear with which name M. Shah should be conflated. In this case, other contextual information needs to be used to resolve the conflict.

5.1.3 Name De-conflation

Our system is also susceptible to unintentionally conflating individuals with the same name. Persons with commonly occurring names such as John Smith, or even Mehul Shah (believe it or not!) will be treated as one. John Smith is likely to be an expert on everything and have many equidistant social relations. The present system does not appear to suffer from this problem because we constrain ourselves to a specific community around a specific person. However, for this method to be successful for the a person with little presence on the web, the name de-conflation must be performed. This issue is more relevant in web pages rather than bibliography collections. If two persons publish with the same name in the same field such artificial intelligence, it is likely that one will often get miscredited for the work of the other. Thus, two persons with identical names often distinguish themselves with an additional name.

One possible approach for de-conflating names is to assume that pages referring to a unique individual are scattered in close proximity to each other with respect to the global hypertext structure. With this assumption, a number of clustering techniques may be employed to sort N documents referring to the same name into M groups. These groupings correspond to documents referring to distinct individuals. An open question is at what threshold to terminate clustering algorithms determining M, the number of distinct persons with a given name.

5.2 Homepages, Hypertext Structure, and Queries

The internet contains information not only explicitly in the actual data that is served, but also implicitly in the nature of the databases and the linkage structure among the documents. Thus far, we have only examined explicit information from document collections; however, these implicit attributes should provide more robust information on the strengths of relations as well as physical and organizational ties of individuals. We describe approaches for harnessing implicit information, thus augmenting the social network data already collected.

In particular, consider homepages and the hypertext structure in which individuals' homepages are connected. Often, homepages contain the strongest evidence of relationships and interests. A mapping of web pages using a service such as Ahoy! to identify pages that are potential homepages may provide an alternative basis for estimating social distance. In this case, a viable distance measure might be a monotonic function of the distance between a pair's homepages. Moreover, web pages often span a connected structure of several pages. Pages connected in such a structure are usually topically related. Hence, "nearby" pages may offer more significant information about a person than all the pages that contain a reference to him.

There are numerous methods for isolating organizational and physical coordinates of individuals. The most obvious method is to identify locations via the URL of the person's homepages. Organizations also have their own set of personal pages for individuals, groups, and projects. Manually identifying and automatically mining social network data from these sources would provide information on the social structure within an institution. With this information, queries may be improved by allowing the user to also specify an organizational constraint.

Currently, we offer only two extreme types of queries for isolating an expert. The localized method is a crude method of traversing a social network for identifying individuals. A best-first search, which was manually simulated in our experiments, might be a more useful algorithm for finding a local expert. The global query essentially extracts names from documents. It might be useful to allow the user to narrow his

search to a reasonable number of documents before performing the the global query.

5.3 Future Evolution

Now we describe two tangential areas of research that are relevant for this prototype to evolve and become practical. First, this system is fundamentally limited since it only relies on public information. Originally, this system set out to address the critical mass and privacy issues that plagued the agent based referral chaining system. Eventually, there will be critical mass of users requiring information about people who do not appear on the web. There will also be an incentive for users to cooperatively join and maintain a profile of themselves on a similar system for finding people. Thus, a hybrid system which combines an agent-based system and our static ReferralWeb is the next logical step. In such a system, an open question is how to handle privacy issues.

The second issue that needs to be addressed is how to effectively scale such a system to millions of users recording not only names but also user profiles. One possible approach is to parallelize the database across many machines and allow searches in parallel. How can a data structure like the social network graph be efficiently distributed among many a cluster of workstations to allowing browsing and visualization?

In this thesis we have described the design and implementation of a prototype for isolating experts and a referral chain by which to contact them. Through some simple experiments we have shown that this system fulfills a need not addressed by any other service. We have also outlined its numerous flaws and possible directions for improvement. Finally, we have described the possible evolution of this system.

Bibliography

[1] Balabanovic, Marko and Shoham, Yoav. (March 1997) "Fab: Content-Based, Collaborative Recommendation," *Communication of ACM'* vol: 40, no: 3, pp. 66-72.

[2] Frakes, W.B. and R. Baeza-Yates, Eds. *Information Retrieval, Data Structures & Algorithms*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[3] Granovetter, M. (1993) "Strength of Weak Ties," *American Journal of Sociology*, vol: 78, pp. 1360-1380.

[4] Krager, D. R. and Stein, Lynn A. *Haystack: Apersonalized IR System*. MIT.

[5] Kautz, H., Selman, B., and Coen, M. (1994) "Bottom-up Design of Software Agents," *Communications of the ACM*, vol: 37, no: 7, pp. 143-146

[6] Kautz, H., Selman, B., and Milewski, A. (1996) "Agent Amplifed Communication," *Proceedings of AAAI-96*, (Portland, Oreg.). Cambridge, MA: MIT Press, pp. 3-9.

[7] Kautz, H., Selman, B., and Shah, M. (March 1997) "ReferralWeb: Combining Social Networks and Collaborative Filtering," *Communications of the ACM*. vol: 40, no: 3, pp. 63-65.

[8] Konstan, Joseph A. et. al. (March 1997) "GroupLens: Applying Collaborative Filtering to Usenet News," *Communications of ACM*. vol: 40, no: 3, pp. 77-87.

[9] Kraut, R., Galegher, and Edigo C. *Intellectual Teamwork: Social and Technological Bases for Cooperative Work*. Hillsdale, NJ: Erlbaum Associates, 1990.

[10] Pattison, P. *Algebraic Models for Social Networks*. New York, NY: Cambridge University Press, 1993.

[11] Rucker, James and Polanco, Marcos J. (March 1997) "Siteseer: Personalized Navigation for the Web," *Communications of ACM*, vol: 40, no: 3, pp. 73-75.

[12] Schwartz, M. F. and Wood, D. C. M. (1993) "Discovering shared interests using graph analysis," *Communications of ACM*, vol: 36, no: 8, pp. 78-89.

[13] Sundheim, B. and Grishman, R., Eds. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. San Francisco, CA: Morgan Kaufmann, 1995.

[14] Terveen, Loren et al. (March 1997) "PHOAKS: A System for Sharing Recommendations," *Communications of ACM*. vol: 40, no: 3, pp. 59-62.

[15] Wasserman, S. and Galaskiewicz, J., Eds. *Advances in Social Network Analysis*. Thousand Oaks, CA: Sage Publications, 1994.

[16] Yao, Y. Y. (1995) "Measuring Retrieval Effectiveness Based on User Preference of Documents," *Journal of the American Society for Information Science*. vol: 46, no: 2, pp. 133-145.'

[17] http://www.altavista.digital.com

[18] http://ahoy.cs.washington.edu:6060

[19] http://liinwww.iro.uka.de/bibliography/index.html

[20] http://www.firefly.com

[21] http://www.four11.com

[22] http://www.hotbot.com

[23] http://www.informatik.uni-trier.de/ ley/db

[24] http://www.infospace.com

[25] http://www.lawcrawler.com

[26] http://www.metacrawler.com

[27] http://www.sixdegrees.com