A Multiagent Referral System for Expertise Location

Bin Yu, Mahadevan Venkatraman, and Munindar P. Singh
Department of Computer Science
North Carolina State University
446 EGRC, 1010 Main Campus Dr
Raleigh, NC 27695-7534
{byu, mvenkat, mpsingh}@eos.ncsu.edu

Abstract

A referral system is an agent-based framework for assisting and simplifying person-to-person communication, such as finding experts for a specified topic. The informal person-to-person social networks are used as "referral chain" requests for expertise, and software agents help automate the search of information and expertise. However, current systems rely exclusively on collections of email messages or newsgroup postings. As a result, previous approaches yield poor referrals for requests outside of the domain of the given collection. By contrast, we develop an approach that (1) combines techniques from information retrieval, multiagent learning and adaptive user modeling to refine the social network according to the user's needs, and (2) allows agents to unintrusively exchange explicit profile information to add coverage regarding a user's expertise. An experimental simulation has been implemented and results of its use are presented.

1 Introduction

There are several ways of accessing information stored online, such as retrieving records from a database or browsing the web. Usually such ways are slow and frustrating. Sometimes we even don't know where to look, despite the various search engines available. On the other hand, much information is simply not online, but only exists as private knowledge. For example, suppose that you need to have your car repaired, and want to know if there is a mechanic with a good reputation near where you live. It is highly unlikely you will find information about reputations from the web or from directories. The most efficient way is to call friends who have had their cars repaired recently or are knowledgeable about mechanics.

Thus, many important kinds of information can only be obtained by "asking a person." Research on social networks also shows that interpersonal communication acts as an important channel for gathering and disseminating information (Katz & Lazarsfeld 1955). But in this scenario the problem of determining the right person to ask remains. E-mail seems to offer a solution to this problem: just mail the problem to everyone who you think might know the answer. But broadcasting emails quickly becomes obnoxious and others will soon begin to ignore such requests (Kautz, Selman, & Milewski 1996).

Many solutions have been developed to access the information online, such as softbots and information agents (Etzioni 1996; Knoblock & Ambite 1997). But few of them actually use the knowledge of interpersonal relationships upon which people rely in daily life. Referral systems integrate online and person-to-person information access by software agents. The informal person-to-person social networks are used as "referral chain" requests for expertise, and software agents help automate the search of information and expertise (Kautz, Selman, & Milewski 1996).

Referral systems have been known for a long time, e.g., (Huhns *et al.* 1987), but their modern incarnations appeared only recently, e.g., ReferralWeb (Kautz, Selman, & Shah 1997), ContactFinder (Krulwich & Burkey 1996), and Knowwho (Kanfer, Sweet, & Schlosser 1997). The agent of each user locates an expert by finding a referral path using available electronic email messages or other information.

One of main problems in current referral systems is the low accuracy of referral chains. We believe it is worthwhile to construct a multiagent referral system to address this problem. In this paper we introduce a two-pronged solution: (1) introduce information retrieval (IR) techniques into multiagent referral systems; (2) allow agents to exchange profile information during interaction. Our purpose is to build an accurate, dynamic and evolving multiagent system that embodies informal social networks that exist in an organization or community.

Section 2 discusses the development and problems of previous referral systems. Our framework and simulation results will be given in section 3 and section 4. Section 5 concludes our paper with a discussion of directions for future research.

2 Referral Systems

In *agent amplified communication*, each agent can access to the following kinds of database files (Kautz, Selman, & Milewski 1996):

- A user-contacts file containing a list of the user's colleagues along with a list of keywords describing their areas of expertise.
- An indexed email file that stores each word that appears in any electronic message along with a list of the messages that contain the word and who sent them.
- A user-profile file containing a list of keywords that describe the user's own areas of expertise.
- A file with names of "close colleagues."

The user formulates a query simply by giving one or more relevant keywords. The agent then scans the user-contacts file and the indexed- email files, and returns to the user a list of names ordered according to the frequency with which the keywords were mentioned in their email correspondence. When the user selects some of the names on the list, the query is passed to the corresponding users' agents. When a user's agent receives a request for expertise, it tries to match the request against its owner's data files. If there is a good match, the request is passed on to its owner directly. If not, the user's agent generates a list of possible referrals using the email records and the user-contact file. If the request originates from someone on the list of close colleagues, this list is passed back automatically to the requesting user's agent. The originator's agent collects all possible referrals, and can continue the process by contacting some of the suggested referrals.

However, Kautz et al. became discouraged after a preliminary test with 25 users suggested that the referrals generated by the agent system were less accurate than those the users would make themselves. Kautz et al. then attempted a similar agent called *ReferralWeb* in which cooccurrence of names in close proximity on World Wide Web pages is used to suggest direct person-to-person relationship. However, it is questionable that a co-citation matrix accurately reflects similar interests. Moreover, ReferralWeb does not take advantage of the person-to-person communication data available from daily email exchanges, which is a common record of communication.

Several other referral systems appeared recently. One is *ContactFinder* proposed by Krulwich and Burkey. A *ContactFinder* is an agent that reads messages posted on bulletin boards, extracting topic areas using heuristics. ContactFinder posts a referral to a person when it encounters a question that matches that person's previous postings. In ContactFinder persons are identified by their postings, and hence that person's communication partners are not considered.

The *Knowwho* email agent maps the user's social network by reading through his or her email messages (Kanfer, Sweet, & Schlosser 1997). It then determines who is best suited to answer the user's question. Email, however, is not always a good indicator of expertise, but rather of interest or experience. We propose using colleague's profile information as a supplement, which can be obtained during the interaction between agents.

3 Technical Approach

The Vector Space Model (VSM)—one of the most famous IR techniques—has proved to be highly effective for many types of documents (Salton & McGill 1983). We adapt VSM to locate people rather than documents. In our formulation, VSM estimates the importance of each term in a query and the term's power of discrimination among the email messages received by the user.

VSM represents queries and documents as vectors in an n-dimensional information space. Then it systematically compares each document vector with the query vector to find the documents nearest to the query. To adapt VSM for referral systems, we represent each person with a vector based on a "document" consisting of all email messages they sent to the user. A person vector contains weights for each term in the query based on how frequently the term was used by the sender in his or her email messages.

Let P_i be the person vector for colleague pi. This vector contains the computed weights for all terms in the sender's email that also occurs in the query vector $\langle w_{i1}, w_{i2}, \ldots, w_{in} \rangle$. The equation for calculating the weights in the person vector for terms 1 to m as:

$$w_{ik} = \frac{tf_{ik} \log(N/n_k)}{\sqrt{\sum_{t=1}^{n} [(tf_{ik}) \log(N/n_k)]^2}}$$

Here, N = number of colleagues of the user, n_k = number of colleagues who have used term k, tf_{ik} = term frequency; number of times colleague p_i has used term k.

The calculation of the query vector, Q_j, can be taken directly from information retrieval methods with computed weights for each term in the query vector $\langle w_{j1}, w_{j2}, \ldots, w_{jn} \rangle$. Salton and Buckley propose several different query vectors. Here is an equation for query term weights that has been shown to be particularly useful for short queries:

$$w_{ik} = [0.5 + \frac{0.5 tf_{ik}}{tf_{max}}] \log(N/n_k)$$

Here N, n_k, and tf_{ik} are the same as above, and tf_{max} = maximum frequency of all terms in query j.

After person vectors have been calculated for each sender, the similarity between the query vector and a person vector can be calculated as the cosine of the angle between those vectors. Given a query vector $Q_j = \langle w_{j1}, w_{j2}, \ldots, w_{jn} \rangle$ and a person vector $P_i = \langle w_{i1}, w_{i2}, \ldots, w_{in} \rangle$, the similarity between the query and a person can be calculated by:

$$sim(Q_i, P_i) = \frac{\sum_{t=1}^{n} w_{jt} w_{it}}{\sqrt{\sum_{t=1}^{n} (w_{jt})^2 \sum_{t=1}^{n} (w_{it})^2}}$$

Using this sort of similarity measure, the agent will return to the user a list of "potential experts" ranked in decreasing order of computed similarity values upon a query. The term weights can be normalized, so the similarity is between 0 and 1. It is convenient to think of this value as the likelihood or confidence that a person will be able to answer the query of the user.

The user may be allowed to specify an absolute relevance threshold. The threshold can be adjusted to tune the number of potential experts found.

Definition 1 Given a query vector Q and a threshold θ, a person vector P is relevant to Q if $sim(Q,P) \geq \theta$.

A colleague's profile is denoted by a profile vector $M_k = \langle w_{k1}, w_{k2}, \ldots, w_{kn} \rangle$. Given a query Q_j and a profile vector M_i, the similarity is calculated by $sim(Q_j, M_k)$. After normalizing each vector to unit length, we can define the relevance of a query to a person.

Definition 2 The relevance of a query Q_j to a person p_i is defined as $Rel(Q_j, p_i) = \max(sim(Q_j, M_i), sim(Q_j, P_i))$, where M_i and P_i are the profile and person vector of the person p_i, respectively.

Each agent can access the following files:

1. A user-profile file containing a user-profile, which describes the user's own areas of expertise
2. A user-contacts file containing a list of the user's colleagues along with descriptions of their areas of expertise. For each colleague p_i, the agent maintains a
   - person vector, which is created based on the electronic messages from p_i
   - colleague-profile vector, which is learned during interactions with p_i
   - score, which depends on how many good referrals received from p_i.

4 Experimental Evaluation and Results

We have completed two main kinds of experiments; other variants are in progress. In our setup, each agent has an *interest* vector, an *expertise* vector, and a *neighbor* model. In general, the neighbor model depends on how many agents know the given agent, how many agents he knows, which community he belong to, and so on. In our case, the neighbor model is this agent's representation of the other agents' expertise.

An agent's queries are generated based on his interest vector. For example, if agent 0 has an interest vector: $[0.0, 0.2, 0.0, 0.0, 0.8]$, then queries from agent 0 are generated by perturbing each element of the interest vector. The motivation for this is to capture the intuition that an agent will produce queries depending on his interests.

When another agent receives a query, he will try to answer it based on his expertise vector, or refer to other agents he knows. Agent 0 collects all possible referrals, and continues the process by contacting some of the suggested referrals. At the same time, he changes his neighbor models for other agents.

Our experiment involves 20 agents with interest and expertise vectors of dimension 5. The agents send queries, referrals, and responses to one another, all the while learning about each others' interest and expertise vectors. The agents are limited in the number of neighbors they may have—in our case the limit is 4. Periodically, they decide which neighbors to keep based on the benefit they have drawn from other agents in terms of responses and referrals.

The effectiveness of a social network is can be defined in terms of the likelihood of obtaining correct answers with the least number of messages. This leads us to define the following metric for the quality of a social network.

Definition 3 The quality of a social network is given by $\sum_{i,j} \frac{sim(I_i, E_j)}{2^{\text{geodesic}(i,j)}}$.

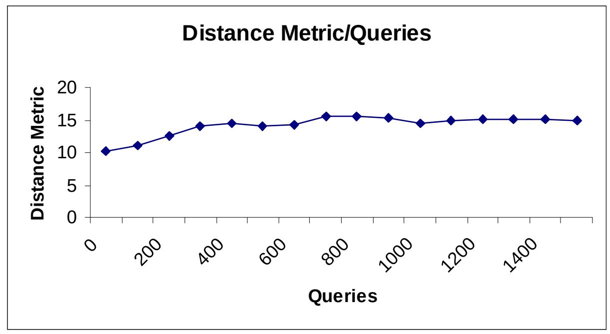

Figure 1: Improvement in the social network

We ran our experiment with interest and expertise vectors assigned randomly to agents. Each agent was given 4 randomly chosen neighbors. Through referrals and learning, the agents found neighbors whose expertise matched their interests (leading to quicker and more accurate responses). Figure 1 shows that the quality metric monotonically decreases over time. This supports the following conclusions.

Result 1 The social network improves over time.

Result 2 The social network is stable under our simulation.

In our second experiment, we carried out a "sensitivity analysis" of the social network. We started with a stable network and injected a new agent randomly into it. The new agent is given random neighbors. Gradually, through referrals, it finds neighbors whose expertise is best related to its interests.

To make this experiment more realistic, we being with a well-known social network, namely, the network of friendship among high-tech managers, which was studied by Krackhardt (Wasserman & Faust 1994). Figure 2 displays Krackhardt's network. In this figure, each node corresponds to a high-tech manager and the edges between them correspond to the managers' friendship with other managers. The edge lengths are

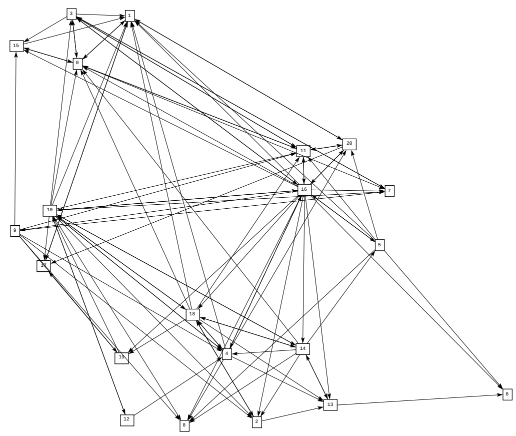

Figure 2: Social network of Krackhardt's high-tech managers

inversely proportional to the extent of friendship (i.e., close friends have shorter edges).

In this experiment, the queries are generated only by the new agent. We found that the quality of the social network from the perspective of the newly introduced agent also improved monotonically and converged.

Result 3 A new person (agent) trying to embed into a social network will drift toward his own community.

5 Conclusions

One reason to believe that referral systems would be useful is that it basically models the manner in which expertise location actually works, while allowing more people to be contacted without causing unnecessary disturbance.

Typically a user is only aware of a portion of the social network to which he or she belongs. Referral systems of the sort developed here not only help a user find experts, but also help the user bridge the gap typically found between communities. A single agent at the intersection of two communities can rapidly help them refer to each other. By evolving into a larger community, the user can discover connections to people and information that would originally lay hidden over the horizon.

The present approach to referral networks is not only useful for building social networks of humans, but we expect can also be applied in building multiagent systems in general. The conventional way to implementing a multiagent system is to use specialized agents such as brokers or facilitators (Huhns & Singh 1998). A referral system approach, being perfectly decentralized, would not only be more resistant to failure but would also lead to the dissemination of better vetted information, leading to superior performance across the system.

References

Etzioni, O. 1996. Moving up the information food chain: Deploying softbots on the web. In *Proceedings of the National Conference on Artificial Intelligence*, 1322–1326.

Huhns, M. N., and Singh, M. P. 1998. All agents are not created equal. *IEEE Internet Computing* 2(3):94–96. Instance of the column *Agents on the Web*.

Huhns, M. N.; Mukhopadhyay, U.; Stephens, L. M.; and Bonnell, R. D. 1987. DAI for document retrieval: The MINDS project. In Huhns, M. N., ed., *Distributed Artificial Intelligence*. London: Pitman/Morgan Kaufmann. 249–283.

Kanfer, A.; Sweet, J.; and Schlosser, A. E. 1997. Humanizing the net: Social navigation with a "know-who" email agent. In *Proceedings of the 3rd Conference on Human Factors and the Web*.

Katz, E., and Lazarsfeld, P. F., eds. 1955. *Personal Influence: The Part Played by People in the Flow of Mass Communications*. Glencoe, IL: Free Press.

Kautz, H.; Selman, B.; and Milewski, A. 1996. Agent amplified communication. In *Proceedings of the National Conference on Artificial Intelligence*, 3–9.

Kautz, H.; Selman, B.; and Shah, M. 1997. ReferralWeb: Combining social networks and collaborative filtering. *Communications of the ACM* 40(3):63–65.

Knoblock, C., and Ambite, L. 1997. Agents for information gathering. In Bradshaw, J., ed., *Software Agents*. Menlo Park, CA: AAAI/MIT Press.

Krulwich, B., and Burkey, C. 1996. The ContactFinder: Answering bulletin board questions with referrals. In *Proceedings of the National Conference on Artificial Intelligence*, 10–15.

Salton, G., and McGill, M. J. 1983. *An Introduction to Modern Information Retrieval*. New York: McGraw-Hill.

Wasserman, S., and Faust, K. 1994. *Social Network Analysis*. New York: Cambridge University Press.