

An Adaptive Social Network for Information Access: Theoretical and Experimental Results*

Bin Yu,[†] Mahadevan Venkatraman, and Munindar P. Singh
Department of Computer Science
North Carolina State University

Abstract

We consider a *social network* of software agents who assist each other in helping their users find information. Unlike in most previous approaches, our architecture is fully distributed and includes agents who preserve the privacy and autonomy of their users. These agents learn models of each other in terms of *expertise* (ability to produce correct domain answers) and *sociability* (ability to produce accurate referrals). We study our framework experimentally to study how the social network evolves. Specifically, we find that under our multiagent learning heuristic, the quality of the network improves with interactions; the quality is maximized when both expertise and sociability are considered; pivot agents further improve the quality of the network and have a catalytic effect on its quality even if they are ultimately removed. Moreover, the quality of the network improves when clustering decreases, reflecting the intuition that you need to talk to people outside your close circle to get the best information.

1 Introduction

There are several ways of accessing information, such as searching the World Wide Web using web search engines or retrieving records from a database. Usually these tools are quite effective if the information exists on-line and happens to be indexed by the databases. However, some information might not be available on-line or might be of a context-sensitive nature that the creator

*This research was supported by Ericsson and by the National Science Foundation under grant IIS-9624425 (Career Award) and ITR-0081742.

[†]Address correspondence to Bin Yu, Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7535, USA. E-mail: byu@eos.ncsu.edu.

of it needs to know who is querying before being able to answer. For example, you might not be able to find a good mechanic by just looking in a telephone directory and you probably wouldn't find detailed evaluations of bad mechanics on-line. The most efficient way is to call friends who have had their cars repaired recently or are knowledgeable about local auto mechanics. Of course, sometimes you might find such evaluations of people and services from some websites such as the BBB or better business bureau, which consolidates ratings information, but you still have no way of knowing who generated those ratings and if the ratings themselves are trustworthy.

Research on social networks has shown that interpersonal communication acts as an important channel for gathering and disseminating information [Katz and Lazarsfeld, 1955, Kautz et al., 1996, Harvey et al., 1998]. In general, one may argue that interpersonal communication is the foundation of social relationships. But if we wish to exploit interpersonal communication for the purpose of finding information, we still need to figure out how to determine the right person of whom to ask a question. There are a couple of obvious approaches that are obviously inappropriate. You cannot just ask all your contacts individually. Further, you cannot hope to stumble upon the right stranger who has the desired information—and even if you did, you wouldn't know how to value the information provided by a stranger.

The importance of referrals has been known for a long time, e.g., [Lin, 1971, Reynolds and Darden, 1971, Reingen and Kernan, 1986], but the ideas of referral systems appeared only recently, e.g., MINDS (based on the documents used by each user) [Bonnell et al., 1984, Mukhopadhyay et al., 1986], and ReferralWeb (based on co-citations, i.e., co-occurrence of names on WWW pages) [Kautz et al., 1997b,a]. The informal person-to-person social networks are used to construct “referral chains” of requests for expertise, and software agents help automate the search of information and expertise [Kautz et al., 1996]. MINDS is the earliest agent-based referral system that we know of, but Kautz *et al.* pioneered the methods of graph analysis for referral systems, in which they find someone with the needed information explicitly using referral chains.

In general, their research is giving much attention to the problem of referral information generation, and did not concentrate on the dynamic aspects of referral systems. Most referral systems maintain all of referral information in a centralized database and simply consider referrals directly through path search in a graph. Neither of them studies the dynamics of social structure, especially any emergence or evolution of agent-based social networks due to referrals. We believe a multi-agent referral system that captures the right intuitions can avoid these problems. In our view, the right intuitions are found in the literature dealing with *small-world networks* [Watts and Strogatz, 1998]. Small-world networks are neither fully regular nor fully random, but capture the structure of real-life human organizations. Watts & Strogatz observe that such graphs have clusters (like regular graphs) and short paths (like random graphs). Whereas sociologists and mathematicians merely study the properties of small-world networks, our goal is to develop a multiagent system that essentially converges to a small-world network, while requiring only local, autonomous decisions by its member agents [Yu and Singh, 1999].

The prototype system MARS is being implemented for use in a practical social network. However, in order both to refine our design and to understand the principles underlying social networks, we have conducted several experiments on a simulation of the above setup. Unlike in most previous approaches, our architecture is fully distributed and includes agents who preserve the privacy and

autonomy of their users. These agents learn models of each other in terms of *expertise* (ability to produce correct domain answers), and *sociability* (ability to produce accurate referrals). We study this framework experimentally to study social networks for knowledge sharing or more broadly information access. All the heuristics used in the simulation are as employed in the actual system.

The rest of this paper is organized as follows. The next section presents our framework for a multiagent referral system. This lays the groundwork for the experimental results of the third section. We briefly summarize the relevant literature in section four. The paper concludes with a discussion of the main themes and directions for future research.

2 Framework

In this context, we describe an adaptive social network for information access, which is intended to be an accurate, dynamic, and evolving multiagent system that can achieve the effect of the informal social networks that exist in an organization or community. We distinguish between a user's interests and his expertise. These two aspects are complementary in that a user will query others based on his interest, but answer others' queries based on his expertise.

In our framework each user is associated with a personal agent. The queries by the user are first seen by his agent who decides the potential contacts to whom to send the query. After consultation with the user, the agent sends the query to the agents for other likely people. The agent who receives a query can decide if the query suits its user and let the user see that query. In addition to or instead of just forwarding the query to the user, the receiving agent may respond with referrals to other users. If the receiving agent or user wish they can discard the query and never respond to it in any way.

A query includes the query vector as well as the requester's ID and email address and a limit on the number of referrals requested. A response may include an answer or a referral or both. As mentioned above, it is allowed to omit a response altogether. An answer if given depends on the query and the expertise of the answering agent. An agent answers only if it is reasonably confident of its expertise matching the incoming query. A referral depends on the query and on the referring agent's model of other agents; a referral is given only if the referring agent has some confidence in the relevance of the agent being referred. Therefore, in order to interact successfully with other agents, each agent has the following information:

- The user's own areas of expertise to decide whether or not to accept incoming queries.
- Models of (some) neighbors to decide whether to send them a given query. These models are learned through interactions with others. The model for a colleague contains its address, expertise vector, and a rating of the sociability.

The initial queries by an agent are sent only to its immediate *neighbors*. It must therefore keep models for these neighbors in its *neighbormodels*. In addition, the agent may keep models of other agents. The models of these other agents are considered the agent's *cache*. The models in the cache are learned from interactions with the other agents, e.g., when they ask or answer a query. The cache agents are not directly contacted for a query, but may be contacted if they are referred

to by someone else. By maintaining additional models in the cache, the agent in effect tracks some acquaintances who may later become its close associates.

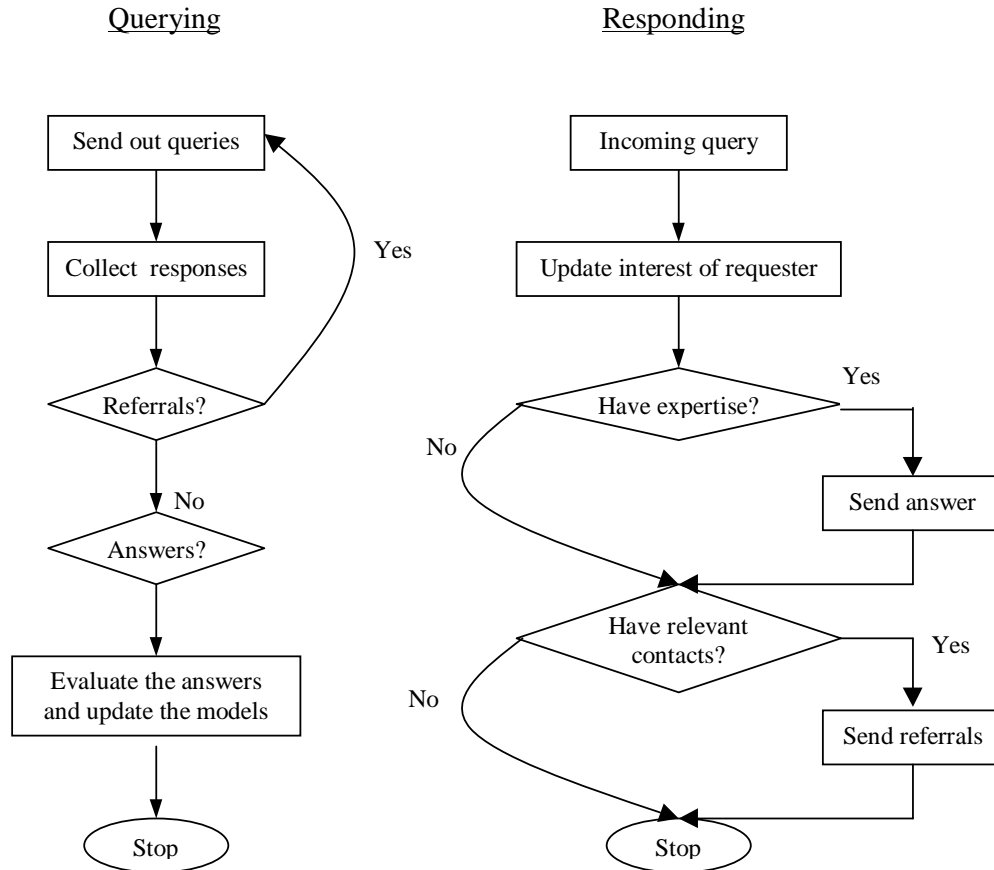


Figure 1: An agent's behavior while querying and responding

When the originating agent receives referrals, it integrates them into its models. Based on its models, it may decide to actually follow up a referral. When the agent receives an answer, it uses the answer as a basis for evaluating the expertise of the agent who gave the answer. This evaluation affects its model of the expertise of the answering agent, and its models of any agents who gave referrals to the answering agent. As a result, if an agent sends in a good answer, the evaluation of its expertise goes up as does the sociability of any agents who referred to it (through any path). Figure 1 sums up the entire process.

2.1 Vector Space Model for Social Networks

The vector space model (VSM) is a classical information retrieval (IR) technique [Salton and 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 users who may be sent that query based on the expertise that they have exhibited. Following the basic VSM idea, we represent the users' interests and expertise (actual and modeled) as vectors in a multidimensional information space. Then we systematically compare each query vector with the expertise vectors of other users to find the user whose expertise is the most similar to the query.

Given an interest and an expertise vector, under VSM, the similarity between the two vectors is defined as the cosine of the angle between those vectors. We give a slightly different definition. This definition too captures the cosine of the angle, but scales it by the length of the expertise vector. Our definition thus captures the projection of the expertise vector on the query vector in the multidimensional vector space. We further normalize this expression by dividing the maximum length possible. The idea is that a person whose expertise for a query is twice as large as another person is twice as desirable as a source of information. Under traditional VSM, two users would be considered equally desirable as long as the angles of their expertise vectors were the same.

Definition 1 Given a query vector $Q = \langle q_1, q_2, \dots, q_n \rangle$ and an expertise vector $E_{P_i} = \langle e_1, e_2, \dots, e_n \rangle$, the similarity between Q and E_{P_i} is defined as:

$$Q \diamond E_{P_i} = \frac{\sum_{t=1}^n q_t e_t}{\sqrt{n \sum_{t=1}^n (q_t)^2}}$$

The query vector is generated from the user's query. The expertise vector depends on each user that the user considers as a potential target of this query. Obviously, the true expertise of the other user may not be known to this user. Therefore, the querying user's agent uses an estimate of the other user's expertise that it has learned from previous interactions. The computed similarity predicts the likelihood that a person will be able to answer the query being posed.

The computation of the above similarity measure identifies the potential known experts for the given query. The agent returns to the user a ranked list of these experts based on similarity. The user can select a few of them to send the query. Each of them may respond with an answer to the query, or referrals to other users, or give no response at all. What the agent needs to learn about others is their potential usefulness for a given query, i.e., the likelihood of getting a good answer or a referral that (in as few steps as possible) yields a good answer.

$Q \triangle p_i$, the relevance of a query Q to a person p_i , is defined as the weighted sum of the expertise and sociability components. The expertise component is essentially the same as above; the sociability component is a scalar, which effectively tells us how to rate the referrals produced by the given person.

Definition 2 Given a query vector Q , the relevance of Q and any neighbor p_j of the user is computed as $Q \triangle p_j = (1 - \eta)(Q \diamond E_{p_i}) + \eta S_{p_j}$, where S_{p_j} is the sociability of the neighbor p_j ; and η and $(1 - \eta)$ are the weights given to sociability and expertise, respectively.

Further, the user may be allowed to specify an absolute relevance threshold. The threshold can be adjusted to tune the number of purported experts found and to limit the number of referrals that this user will give other users.

Definition 3 Given a query vector Q and a threshold Ω , a person p_i is relevant to Q if $Q \triangle p_i \geq \Omega$.

2.2 Multiagent Learning

A *referral graph* encodes how the computation spreads in a social network as a query originates from an agent and referrals or answers are sent back to this agent. Figure 2 shows an example referral graph (with the dashed lines). Here agent *A* asks a question of two of its neighbors *B* and *D*, and *B* leads to a good answer (from *J*). *D* sends back a referral to *F*, which leads to a non-answer. Using the referral graph, it is easy to visualize how the agents learn about each other in our approach.

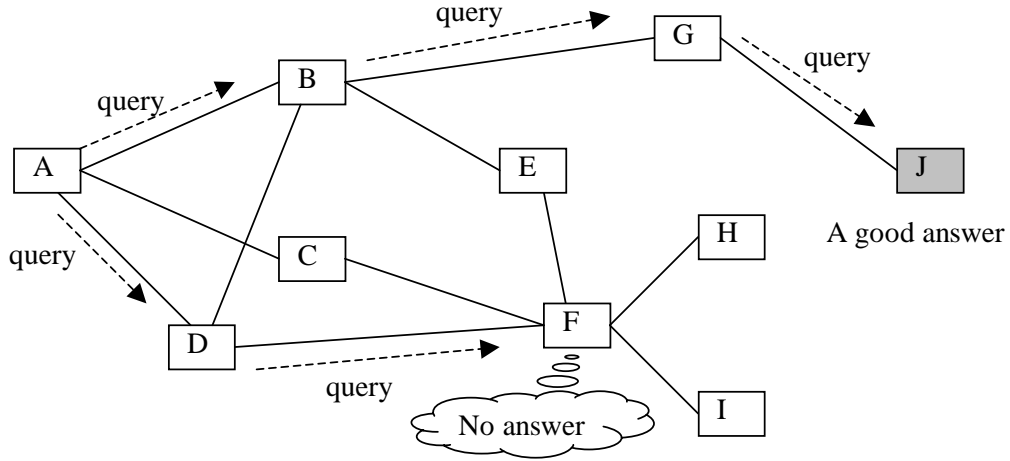


Figure 2: An example referral graph

Formally let Q be a query from user p_i . We assume that, after a series of l referrals, this produces a response R from colleague p_j . Let the entire referral chain in this case be $\langle p_i, r_1, \dots, r_{l-1}, p_j \rangle$. Now we can refine the estimate of the correspondent's relevance towards certain queries, and use that to adapt the models that p_i has of everyone else involved in this process. In the following, let E_Q be the estimated expertise of p_j as is reflected in p_j 's response R to query Q . There are three main possibilities. Here α and β are learning rates; S_k denotes the sociability of referring agent r_k ; B is the branching factor, i.e., the maximum number of referrals that are allowed.

- A good answer. The estimated expertise for p_j is updated as $E_j = (1 - \alpha)E_j + \alpha E_Q$. For all intermediate vertices, the sociability is updated as $S_k = S_k + \frac{(1-S_k)\beta}{B^{(l-k)}}$.
- A bad answer. The estimated expertise for p_j is updated as $E_j = (1 - \alpha)E_j + \alpha E_Q$ (same formula as above, but E_Q is lower in this case). For all intermediate vertices, the sociability is updated as $S_k = S_k - \frac{S_k\beta}{B^{(l-k)}}$.
- No answer. This is treated the same as a bad answer with the estimated expertise set to a low value. That is, for any referral chain which peters out into nothing, all members of the chain are penalized.

The learning is additive with respect to the referral chains. That is, we treat all referral chains one by one in some arbitrary order. If p_j is referred to be many users and produces a good answer, all of them gain in sociability. Conversely, if p_j produces a bad answer, all of them suffer in sociability.

3 Experimental Results

In our simulation, there are no users of course, so we treat the agent and user simply as just the agent. Each agent has an *interest* vector, an *expertise* vector, and models of several neighbors. In general, the neighbor’s model depends on how many agents know the given agent, how many agents it knows, what kinds of agents those are, and so on. In our case, the neighbor’s model kept by an agent are the given agent’s representation of the other agents’ expertise and sociability.

An agent’s queries are generated based on its interest vector. The queries are generated as vectors by perturbing the interest vector of the given agent. The motivation for this is to capture the intuition that an agent will produce queries depending on its interests. When an agent receives a query, it will try to answer the query based on its expertise vector, or refer to other agents it knows. The originating agent collects all possible referrals, and continues the process by contacting some of the suggested referrals. At the same time, it changes its models for its neighbors.

Our experiments involve between 20 and 60 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. The neighbors are the agents to whom a given agent may send a query or about whom the agent may issue a referral. The idea is that the agent should have few neighbors relative to the entire society of agents. This is similar to the intuition that people living in a town of 300,000 may only know, say, 200 people—i.e., a fairly small fraction of the entire set.

In general, the agents may keep track of more neighbors than just their neighbors. Periodically, they decide which neighbors to keep in the neighbor models, i.e., which are worth remembering. They also decide which of the neighbors to promote to neighbors based on the benefit they have drawn from these and other agents in terms of answers and referrals.

3.1 Metrics

The effectiveness of a referral network 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 referral network.

Definition 4 The quality of a referral network is given by the following, where $\text{path}(A_i, A_j)$ gives the shortest path length from agent A_i to principal A_j , N is the total number of agents.

$$\sum_{A_i, A_j} \frac{I_{A_i} \diamond E_{A_j}}{N * \text{path}(A_i, A_j)}, \text{ for any two agents and } A_j \neq A_i$$

As a base-line for comparison, we also introduce the *direct query* metric, where the agents do not seek or grant referrals to anyone.

Definition 5 The direct query metric of a referral network is given by the following, where N is the total number of agents, $Degree$ is the number of neighbors for each agent:

$$\sum_{A_i} \sum_{A_j \in A_i's \text{ neighbors}} \frac{I_{A_i} \diamond E_{A_j}}{N * Degree}$$

3.2 Bootstrapping

Following Watts & Strogatz [Watts and Strogatz, 1998], we begin from a regular ring but allow for edges to be *directed*. we use a regular ring with 100 nodes, and 4 edges per node (to its nearest neighbors) as a start point for the experiment. During each simulation cycle, we randomly designate an agent to the “requester” agent and update the neighbor models of each agent upon the responses. Since we have not real users here, how good of an answer depends on how close of the answer to the interest vector.

There are some assumptions for the experiment. As Kautz *et al.* claimed in [Kautz et al., 1996], when searching for a potential expert agent in the network, usually the further removed a expert is from the requesting agent, the less likely the expert agent will respond. Similarly, the more steps away from the requester, the less accurate of the referrals provided. We set a bound of *six* for the length of any referral chain. Another assumption is that each agent in the experiment has the same thresholds for the relevance of a query and expertise vector for principal himself or neighbors. For each agent A_i , we have $\Omega = 0.1$. The number of neighbors is 4 and the learning rate $\alpha = \beta = 0.1$. An answer is a good answer if and only if similarity value between the query and the answer was above 0.1.

3.3 Stability

It is interesting to study the stability of the referral network under different conditions, i.e., relationship of interest and expertise vectors, the weight of sociability and threshold. In the section we will study the quality of network when the relationship of the interest vector and expertise vector for each agent are: *approximately equal - similar*, *independent*, and *opposite*. We ran our experiment with interest and expertise vectors assigned to agents for some fixed weight of sociability (0.3 in this setting). Each agent was given four randomly chosen neighbors (from a *directed* regular ring). Through referrals and learning, the agents found neighbors whose expertise matched their interests (leading to quicker and more accurate responses).

Figure 3 shows that the quality metric and direct query metric do increases virtually monotonically over time in the beginning when the interest and expertise vectors are opposite, and then stabilize at a high level. We also get similar results for other two cases where the interest and expertise vectors are similar or independent. For the rest of the experiments, we only consider the cases when the interest and expertise vectors are opposite. This situation is quite natural since it captures the case where principals are interested in areas where they have the least expertise.

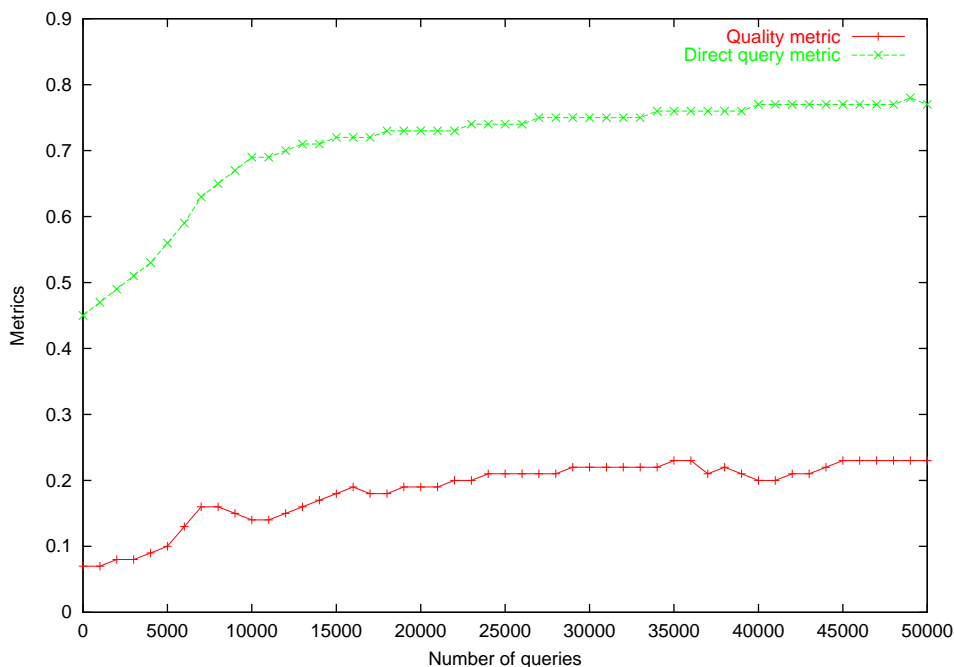


Figure 3: Quality of network when interest and expertise vectors are opposite

3.4 Effect of referrals

Figure 4 shows the effect of referrals on the quality of the network. When the weight of sociability (η) is set of 0, that means referrals have no effect on the learning; agents learn and remember about other agents based solely on their expertise. When the weight of sociability is set to 1, the agents remember other agents based solely on their ability to give referrals, not on their ability to actually answer queries. As is intuitively obvious, neither of these extremes is the best.

Figure 4 confirms that the ability to give and take referrals significantly improves the quality of the network. Note that as η increases, the direct query metric decreases, which shows that there is a trade-off between expertise and sociability, and this is exactly what the quality metric captures. Figure 4 also confirms that a certain amount of emphasis (during learning and querying) on the agents' referring ability improves the quality of the network, but that an overemphasis on referrals to the cost of expertise is not useful. For simplicity, we only consider the cases while $\eta = 0.3$ for the rest of the experiments if not specified.

3.5 Pivot agents

We model pivots as agents who have a significantly higher out-degree (i.e., number of neighbors) than other agents. Because of their higher out-degree, such agents are valuable to others and soon end up with a high in-degree as well. In our simulations we specify agent 0 as the only pivot agent (suppose we have 100 agents all together) and add its out-degree with the probability 0.8. Our simulations confirmed the hypothesis that the existence of a pivot agent significantly improves the

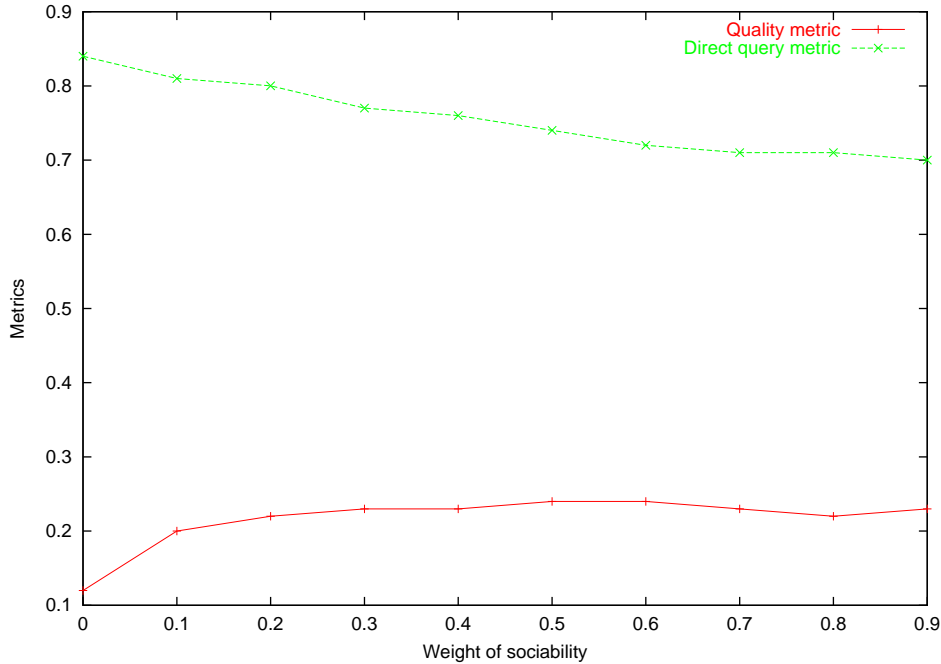


Figure 4: Usefulness of referrals at different weights of sociability (Quality metric)

quality of the referral network as perceived by all agents in the network (Figure 5).

Besides the pivots, we also observe a *catalytic* effect. When we add a pivot agent to a stable system, the system converges to a better quality. This is to be expected because the pivot after all brings in far more contacts and would lead to shorter paths between agents in different communities. When we take out the pivot, the system once again converges to a reduced quality. However, interestingly, the resulting quality is significantly higher than the original level (see Figure 6). We term this net gain in quality the *catalytic effect*. The catalytic effect is powerful, because it is attained simply by rewiring the graph without changing the number of edges in it.

3.6 Sensitivity analysis

Clearly, a referral network will not remain stable for long, because agents will continually introduce and remove themselves from the network. To evaluate how our approach will accommodate changes of this variety, 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.

In this experiment, the queries are generated only by the new agent. We found that the quality of the referral network from the perspective of the newly introduced agent improves as it sends out more queries, and converges to a similar high level as other agents in terms of direct query metric. Figure 7 confirms this hypothesis.

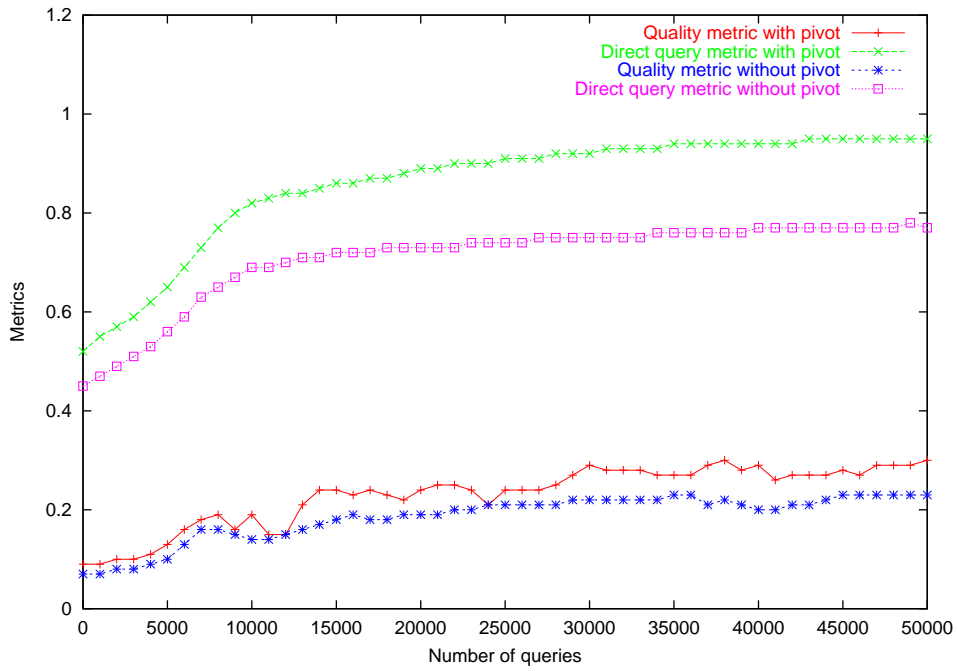


Figure 5: Improvement in quality due to a pivot at different sociability weights

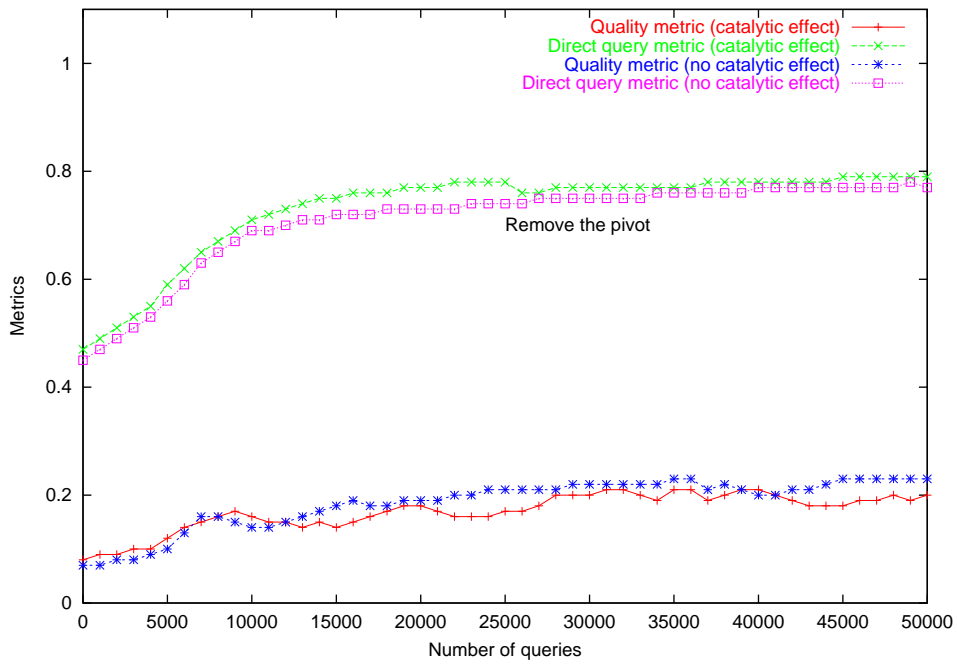


Figure 6: The catalytic effect of a pivot.

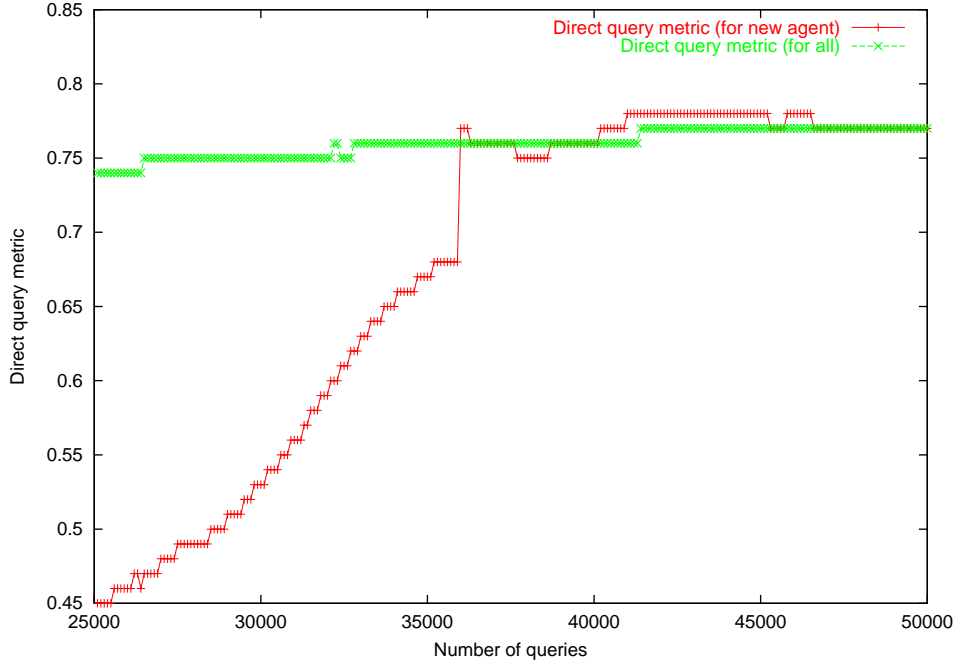


Figure 7: Improvement of quality perceived by a new agent in a stabilized network

3.7 Structure analysis

A *clique* is a maximal complete subgraph of three or more vertices [Wasserman and Faust, 1994]. Intuitively, a clique reflects a cluster in a graph. Watts & Strogatz propose a clustering coefficient to estimate the cliquishness of an undirected graph [Watts and Strogatz, 1998] with the intent of capturing how well the neighbors of a vertex are connected to each other. We define a metric called *reflexive clustering* (C) that seeks to capture a similar intuition for a directed graph. It differs from Watts & Strogatz's clustering in two respects: it applies to a directed graph and it accounts not only for how well the neighbors of a vertex are connected to each other, but also how well the neighbors are connected to the vertex itself.

In the following, let G be a graph with $|G|$ vertices; let v be a vertex in G . Let N_v include all the neighbors of v and v . Let E_v include all the edges among the vertices in N_v that are present in G . (there may be up to $|N_v|(|N_v| - 1)$ such edges).

Definition 6 We define reflexive clustering as (C_G) as follows.

$$C(v) = \frac{|E_v|}{|N_v|(|N_v| - 1)}$$

$$C_G = \frac{\sum_{v \in G} C(v)}{|G|}$$

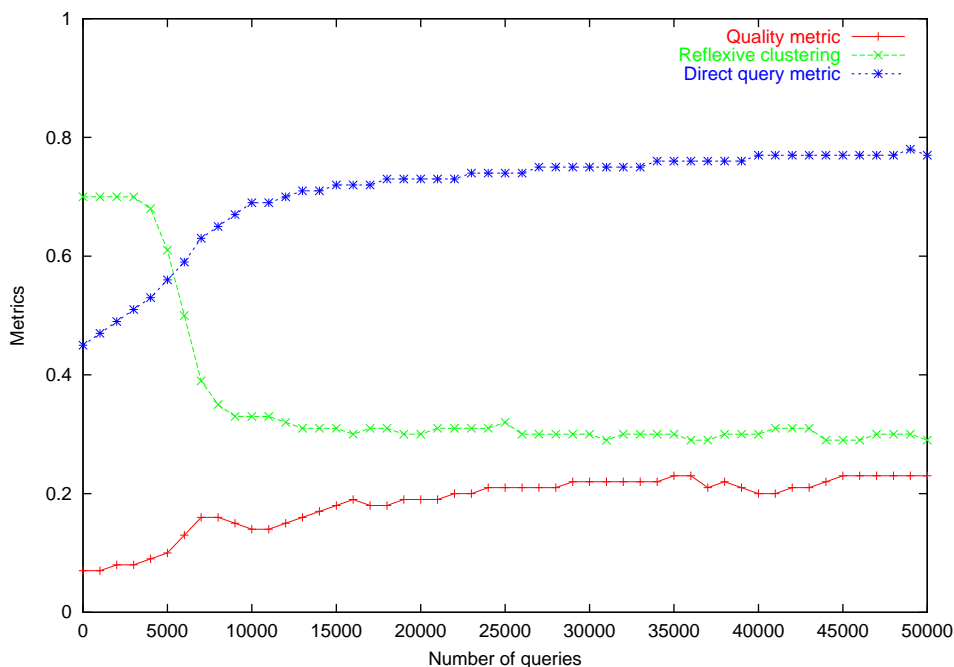


Figure 8: The change of clustering and the quality of the network starting from a regular graph

Note that it does not mean a graph is *connected* when $C_G = 1$. C_G measures the cliquishness of a typical neighborhood (a local property). It could be several graphs and each of them is *complete*. Watts & Strogatz studied the structural properties of small-world networks by rewiring a regular ring lattice with different probabilities. It is more interesting to study the structure change of the whole group after a series of queries, answers, and referrals. We may begin with a regular ring, and see if it will converge to a small-world network according the two metrics we mentioned above. Will it continue to converge to the random graph site, or *oscillate* in the some ranges?

Figure 8 shows the change of clustering when starting with a regular ring. As claimed by Watts & Strogatz, there is a rapid drop of average path length corresponding to the onset of small-world phenomenon. During the drop, C_G remains almost constant at its value for the regular lattice, indicating that the transition to a small-world is almost undetectable at the local level. However, when the users have different interest and expertise, then the people who need to know them are different from the people they need to know. We observed the quality of the social network (for both quality metric and direct query metric) improves while the clustering decrease. Figure 9 shows the change of clustering starting from a random graph. We find reflexive clustering oscillate around 0.4, which tells the transition from a random graph to a small-world network is slow.

4 Related Work

Several other referral systems also appeared around 1996. One is ContactFinder [Krulwich and Burkey, 1996], which is an agent that reads messages posted on bulletin boards, and extracting

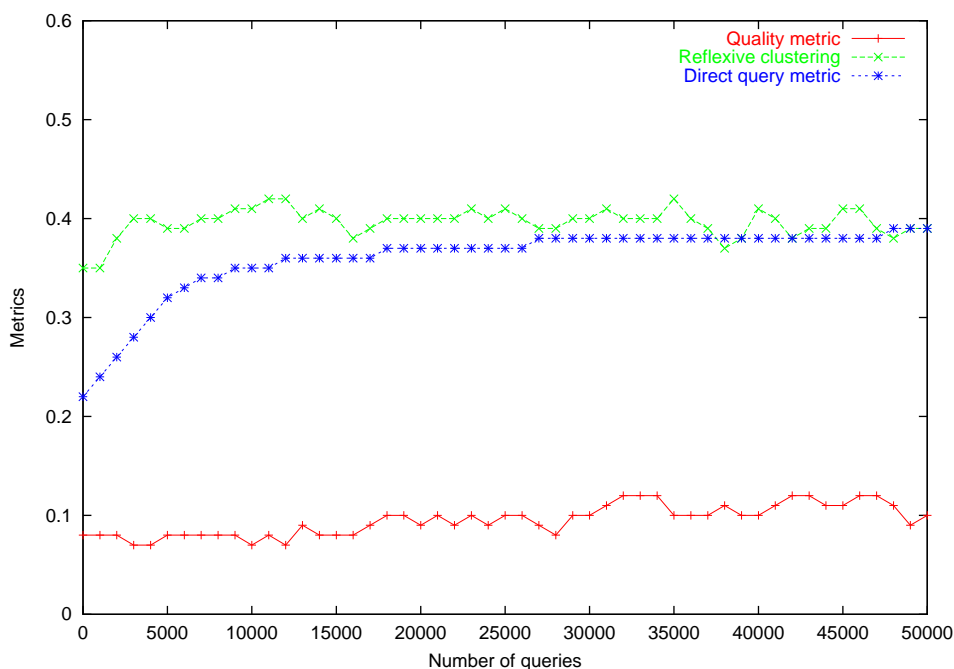


Figure 9: The change of clustering and the quality of the network starting from a random graph

topic areas using a set of heuristics. ContactFinder assists users by referring them to other people who can help them, rather than attempting to find information that directly answers the user’s specific question. However, rather than simply communicating with the users, ContactFinder posts its referrals back to the bulletin boards, and hence that person’s communication partners are not considered.

The Know-Who email agent maps the user’s social network by reading through his or her email messages [Kanfer et al., 1997]. They used three techniques to improve the accuracy of referrals: (1) term weighted document matching methods adapted to locating persons, (2) relevance feedback, and (3) semantic generalization for terms used in queries. They did not consider the sociability of each user, and any strategy of controlling the process of referral chaining.

Other works appeared more recently, which focus on how to mine the structure of social networks by analyzing Java programs, or publications and activities in an organization. Vivacqua *et al.* developed a user-interface agent, called Expert Finder [Vivacqua and Lieberman, 2000], which can assist a novice user in finding experts to answer a question by matchmaking between profiles of the novice and the expert. One of the MITRE projects also called Expert Finder [Maybury et al., 2000] which derives expertise estimation from newsletters, resumes, employee database and other information in an organization. MITRE’s XperNet focuses on identification and tracking of expert communities using statistical clustering and network analysis techniques. Answer Garden [McDonald and Ackerman, 1998, 2000] is a system designed to help in environments such as a help desks. It provides a branching network of diagnostic questions through which experts can navigate to match the novice’s question. Moreover they use some approximation techniques for mapping expertise networks (specializations of an organization’s social network) within an organization.

5 Conclusions

Social networks are a natural way for people to go about seeking information [Gladwell, 1999]. One reason to believe that referral systems would be useful is that they naturally model the manner in which expertise location 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.

These results are interesting in that they suggest that referral systems are the opposite of situations where techniques such as collaborative filtering are best applied [Shardanand and Maes, 1995]. In collaborative filtering, recommendations for a user are generated by a centralized system based on the selections of people deemed most similar to the given user. In such a setting, clustering the users appears to be a good idea. By contrast, here we find that when the recommendations or referrals are generated in a distributed manner, the quality of the network improves when clusters is reduced. This corresponds to the intuition that people benefit from knowing others outside their parochial groups. We don't claim that random scattering is useful, but that some distribution over communities is helpful.

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 [Decker et al., 1997, Huhns and 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.

In future work, we plan to formulate and study the right notion of scattering and look for optimized ways for a social network to attain such scattering. The catalytic effect suggests that our general approach does not find the best networks, because the quality can be easily improved by adding a pivot agent. Can the quality be improved even further? Another of our tasks is to study refined heuristics to improve the quality of the network.

References

- Ronald Bonnell, Michael Huhns, Larry Stephens, and Uttam Mukhopadhyay. MINDS: Multiple intelligent node document servers. In *Proceedings of IEEE First International Conference on Office Automation*, pages 125–136, 1984.
- Keith Decker, Katia Sycara, and Mike Williamson. Middle-agents for the internet. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 578–583, 1997.
- Malcolm Gladwell. Six degrees of Lois Weisberg. *New Yorker*, pages 52–63, January 1999. January 11 issue.
- Clare F. Harvey, Peter Smith, and Peter Lund. Providing a networked future for interpersonal

- information retrieval: InfoVine and user modelling. *Interacting with Computers*, 10:195–212, 1998.
- Michael N. Huhns and Munindar P. Singh. All agents are not created equal. *IEEE Internet Computing*, 2(3):94–96, June 1998. Instance of the column *Agents on the Web*.
- Alaina Kanfer, Jim Sweet, and Anne E. Schlosser. Humanizing the net: Social navigation with a “know-who” email agent. In *Proceedings of the 3rd Conference on Human Factors and the Web*, 1997.
- Elihu Katz and Paul F. Lazarsfeld, editors. *Personal Influence: The Part Played by People in the Flow of Mass Communications*. Free Press, New York, 1955.
- Henry Kautz, Bart Selman, and Al Milewski. Agent amplified communication. In *Proceedings of the National Conference on Artificial Intelligence*, pages 3–9, 1996.
- Henry Kautz, Bart Selman, and Mehul Shah. The hidden web. *AI Magazine*, 18(2):27–36, 1997a.
- Henry Kautz, Bart Selman, and Mehul Shah. ReferralWeb: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, March 1997b.
- Bruce Krulwich and Chad Burkey. The ContactFinder: Answering bulletin board questions with referrals. In *Proceedings of the National Conference on Artificial Intelligence*, pages 10–15, 1996.
- Nan Lin. Information flow, influence flow and the decision-making process. *Journalism Quarterly*, 41:33–40, 1971.
- Mark Maybury, Ray D’Amore, and David House. Automating the finding of experts. *Research Technology Management*, 43(6):12–15, 2000.
- David W. McDonald and Mark S. Ackerman. Just talk to me: A field study of expertise location. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW)*, pages 315–324, 1998.
- David W. McDonald and Mark S. Ackerman. Expertise recommender: A flexible recommendation architecture. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW)*, pages 231–240, 2000.
- Uttam Mukhopadhyay, Larry Stephens, Michael Huhns, and Ronald Bonnell. An intelligent system for document retrieval in distributed office environments. *Journal of American Society for Information Sciences*, 37:123–135, 1986.
- Peter H. Reingen and Jerome B. Kernan. Analysis of referral networks in marketing: Methods and illustration. *Journal of Marketing Research*, 23:370–378, November 1986.

- Fred D. Reynolds and William R. Darden. Mutually adaptive effects of interpersonal communication. *Journal of Marketing Research*, 13:449–454, 1971.
- Gerald Salton and M. McGill. *An Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.
- Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, pages 210–217, 1995.
- Adriana Vivacqua and Henry Lieberman. Agents to assist in finding help. In *Proceedings of ACM CHI'00 Conference on Human Factors in Computing Systems*, pages 65–72, 2000.
- Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, New York, 1994.
- Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, June 1998.
- Bin Yu and Munindar P. Singh. A multiagent referral system for expertise location. In *Working Notes of the AAAI Workshop on Intelligent Information Systems*, pages 66–69, 1999.