Trust Based Knowledge Outsourcing for Semantic Web Agents *

Li Ding
Department of CSEE
UMBC
Baltimore MD,USA
dingli1@umbc.edu

Lina Zhou
Department of Information Systems
UMBC
Baltimore MD, USA
zhoul@umbc.edu

Timothy Finin
Department of CSEE
UMBC
Baltimore, MD,USA
finin@umbc.edu

Abstract

The Semantic Web enables intelligent agents to "outsource" knowledge, extending and enhancing their limited knowledge bases. An open question is how agents can efficiently and effectively access the vast knowledge on the inherently open and dynamic Semantic Web. The problem is not that of finding *a source for desired information, but deciding which among many possibly inconsistent sources is most reliable. We propose an approach to agent knowledge outsourcing inspired by the use trust in human society. Trust is a type of social knowledge and encodes evaluations about which agents can be taken as reliable sources of information or services. We focus on two important practical issues: learning trust and justifying trust. An agent can learn trust relationships by reasoning about its direct interactions with other agents and about public or private reputation information, i.e., the aggregate trust evaluations of other agents. We use the term trust justification to describe the process in which an agent integrates the beliefs of other agents, trust information, and its own beliefs to update its trust model. We describe the results of simulation experiments of the use and evolution of trust in multi-agent systems. Our experiments demonstrate that the use of explicit trust knowledge can significantly improve knowledge outsourcing performance. We also describe a collaborative trust justification technique that focuses on reducing search complexity, handling inconsistent knowledge, and avoiding error propagation.*

1. Introduction

In a human society, a person learns part of his knowledge from direct observations and experiences. Much more is acquired through interactions with other people, in the form of volunteered assertions, indirect implications, and answers to queries. Individuals often decide to rely on other individuals as their source of a particular class of facts, e.g., I use the channel 5 weatherman as a reliable source of basic facts about some meteorological facts. A similar phenomenon occurs in many multi-agent systems. An agent has direct access to some information sources (e.g., databases or sensors) and receives other information through communication with other agents. Moreover, some agents are designed to serve the role of providing certain classes of facts to others. We refer to this situation – that one agent commits to using another as its source of information for a class of facts – as *knowledge outsourcing.*

The emerging Semantic Web, which integrates logical inference, knowledge representation, and intelligent software agent technologies [3, 16], introduces the notion of a *Semantic Web agent* that is intelligent and capable of outsourcing knowledge resulting in a distributed agent society. Knowledge outsourcing in Semantic Web agent society consists of three key steps: understanding, search, and justification. *Understanding* refers to how an agent interprets and understands the knowledge obtained from other agents. *Search* pertains to how an agent discovers and selects candidate knowledge sources. *Justification* concerns about how an agent evaluates and justifies the truthfulness and reliability of acquired knowledge.

Most current semantic web research has focused on the first of these three steps. The use of explicit ontologies described in well defined knowledge-based markup languages like OWL is an approach that directly supports understanding. The remaining steps, search and justification, are challenging given the open and dynamic nature of the web and are the subject of current research.

Trust is an important aspect of social interactions in human society and and potentially in agent soci-

*This research was supported in part by DARPA contract F30602-97-1-0215.

eties as well. Inspired by Marsh [21], we proposed a approach that uses trust as social heuristic to handle the complexity and uncertainty involved in knowledge outsourcing in Semantic Web agent society.

This paper is organized as following: section two discusses trust in a knowledge outsourcing context; section three presents practical trust learning methods and a trust based justification algorithm; section four analyzes experimental results; section five reviews related work; and section six concludes our work.

2. Trust in knowledge outsourcing

Trust has been extensively studied from multiple disciplinary perspectives, including sociology, e-commerce, multi-agent systems, and security [22, 13, 15, 6]. The Semantic Web brings new opportunities and challenges to trust research. On one hand, the machine-understandable knowledge resources freely available on the Semantic Web provide unparalleled infrastructure for building trust relationship. On the other hand, the vast amount of information, including inconsistent and contradictory information, demands trust to address the many practical concerns.

We begin by outlining characteristic features of trust drawing on work by Marsh [22], Grandison and Sloman [15] and continue by focusing on the Semantic Web knowledge outsourcing problem. Significant trust features include the following.

1. *Trust is context dependent.* Trust has different meaning in different context, e.g. in medical context, trust is about the trustworthiness of medical knowledge.

2. *Trust is subjective.* Trust is the social knowledge that is derived from personal observations and serves for future personal decision-making.

3. *Trust is an asymmetric, binary relation.* A trust relation has a trustor and a trustee. It is asymmetric such that "A trust B" doesn't necessarily imply "B trust A". Moreover, both trustor and trustee could be one or many agents.

4. *Trust has value.* A trust relation is associated with a value that represents it's strength or degree of truth. The form and meaning of the value varies, depending on the approach and use. Common examples include boolean values, multi-valued logics, fuzzy logic values, probabilities, discrete rankings, etc.

5. *Trust is conditionally transitive.* Trust in security sense is always intransitive, but in recommendation sense it is partially transitive.

6. *Trust has a temporal dimension.* Since trust is learned from past observations, trust values evolve with new observation and experience. Moreover, to account for changes in a trustee's behavior, recent observations carry more weight in deriving trust.

2.1 Trust ontology

A trust ontology starts from the taxonomy of an agent's knowledge. Consider a scenario in which a patient visits a doctor. The doctor can either directly treat the patient drawing on his own expertise or recommend other specialists. In this case, the doctor uses two types of knowledge: his medical expertise (domain knowledge) and his beliefs about the other specialists' expertise (social knowledge). *Domain knowledge* refers to expertise, and *Social knowledge* covers an agent's beliefs about both the other agents' features and the inter-agent relations such as friend-of, member-of, know, friend, love, and trust. It is similar to the idea of Searle's social facts [26]. Trust is one kind of context dependent social knowledge. In a knowledge outsourcing context, trust is one agent's beliefs about other agents' knowledge. This implies that most agents are cooperative, fair and truthful. In other words, we assume that most agents tell what they believe, they believe what they say, and, in general, behave in a similar way with all other agents.

We propose two types of trust in such context: *domain trust*, which refers to an agent's beliefs about the trustworthiness (or usefulness) of other agents' knowledge in a certain domain, and *referral trust*, which relates to an agent's beliefs about the trustworthiness (or usefulness) of other agents' referral knowledge. Furthermore, we could refine the referral trust into two sub-types: *expert-ref trust*, which is referral trust about experts, and *referrer-ref trust*, which is referral trust about referrers. The suggested knowledge hierarchy is shown in figure 1.

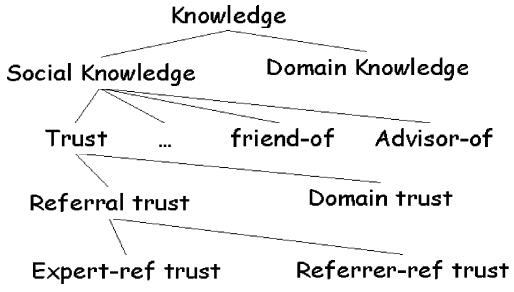

Figure 1. Agent knowledge taxonomy

The trust concept in knowledge outsourcing has

some important properties or slots. (1) The *trustor* slot indicates the agent (or a group of agents) who believes this trust. Normally, the trustor is the agent who holds this piece of trust knowledge. (2) The *trustee* slot indicates the agent or (agents) being trusted. (3) The *domain* slot bounds which part of trustee's knowledge to be trusted. This slot helps trustor find the relevant agents for a query. (4) The *coverage* slot shows how much the trustee knows in that domain and allows the trustor to rank agents by the probability of giving an answer. (5) The *accuracy* slot measures the fidelity of an agents answers. (6) The *reliability* slot shows the confidence that the trustor can get a right answer from the trustee. (7) The *timestamp* slot records the time the trust relation is derived, so as to indicate the freshness of trust.

2.2 Trust value and ignorance

We use the term *ignorance* to refer to an agent being uncertain of or uncommitted to its belief in a sentence. Modeling ignorance is important in trust representation; when considering whether to query an agent, we need to predict whether it will give us the right answer, a wrong answer, or fail to give us any answer. Some researchers [17, 30] suggest using the triplet *(trust, distrust, ignorance)* to grade trust; however, their approaches require prior knowledge about ignorance (the uncommitted belief). Grandison and Sloman [15] observe that it is hard for users to determine the proportion of ignorance in a domain. *We are ignorant of how much we don't know!*

In knowledge outsourcing, in view of the initiator's social experience, an individual participant's knowledge in a domain consists of four parts: right, wrong, unknown, and untouched. The first three are derived from the past experience. Specifically, *right* and *wrong* relate to the experiences where answers were supplied, and *unknown* relates to the experiences where the participant answered with "not known". *Untouched* covers the rest of domain that are not solicited. It is hard to attribute the unknown part to any elements in the triplet (trust, distrust, ignorance) because it reveals the trustee's ignorance. Therefore, we need to expand the triplet into a quadruplet (trust, distrust, unknown, ignorance) when deriving trust value.

2.3 Transitivity of trust

Research communities differ on whether trust should be considered a transitive or intransitive relation. Christianson and Harbison [7] narrowed trust as a highly subjective binary security primitive and argued that trust is intransitive. One of their proofs is unintentional transitivity: if an entity A trust B but does not trust C and entity B trust C, B may add derived trust assertions to A's trust base without A's explicit consent. The notion of intransitive trust is generally accepted by the security community since transitive trust is not always reliable [15].

But we do observe some transitive trust when we loosen the definition of trust, especially in cases that do not require full reliability. Referral trust [1, 8, 31] is partially transitive in a knowledge outsourcing context. The reasons are: (1) referral trust is recursively defined, i.e. referral trust is the trust about the other agents' referral trust; (2) the initiator has full control over consensus query and can thus avoid unintentional transitivity; and (3) the effect of "false" trust can be mitigated through trust refinement and evolution.

2.4 Personal trust and public trust

There are two classes of trust commonly used in agent society: personal trust and public trust. *Personal trust* is derived from an agent's own social experiences (i.e., first hand experiences). It is usually subjective and serves as the basis for the agent's future decisions. *Public trust* is based on the reported social experiences (i.e., second hand experiences) throughout agent society, and it reflects the general opinion about individuals. Normally, public trust, such as reputation system [20, 25, 32], is used as initial trust about unfamiliar agents. Therefore close loop control could be used to keep the agent society productive.

The major difference between personal trust and public trust comes from the social experiences on which they are based. *Social experience* in knowledge outsourcing has some unique features: initiator, participant, query, answer, and reward (the initiator's judgment about the answer). The answer is derived from the tuple of (initiator, participant, query), and the reward is affected by the tuple of (initiator, participant, answer). For personal trust, all related social experiences have similar patterns: same initiator, same participant, and same domain of query. For public trust, it may involve various uncertain factors: initiators may have their own reward mechanisms; a participant may treat initiators differently; and not all social experiences are reported. Though these factors bring difficulties, public trust could still be reliably derived and used because: (1) public trust is statistically reliable when its value is extreme (trust or distrust) and enough experiences have been observed; (2) people tend to report extreme values ("distrust" or "trust") to public; (3) abnormal social experiences can be recorded as valuable

reference for high risk decisions; and (4) public trust can be used to guide the adjustment of personal trust.

2.5 Co-evolution of trust

Trust has an important temporal dimension. Trust management uses relatively static trust: extensive identity authentication needs to be achieved in the first-time experience, and this can be reduced to verifying whether one is dealing with the same party in subsequent transactions. In other words, when the environment and the specific circumstances are safe and reliable, less trust is necessary for reliance. However, the constant diversification and merging of domains and advancement of domain knowledge may invalidate past relationships to some extent. In terms of knowledge outsourcing, trust has a decidedly dynamic characteristic.

After initial trust formation, the trust relationship evolves as additional evidence becomes available. It is more so in the ubiquitous computing environment. Accumulation of evidence with experience of new interactions must modify the level of trust to be placed in an entity, incrementing the summary information to maintain accuracy [11]. In particular, trust updating is a co-evolving process in the trust network. For example, suppose agent A creates initial trust with entity C through entity B. The degree of trust may be adjusted up or down depending on A's interactive experience with C. The adjusted trust may be propagated back to the original trust relationship between B and C.

2.6 Distributed vs. centralized

Traditionally, trust is managed with a centralized approach. One typical example is security technology featuring a trusted third party (TTP) or trusted authority. As the number of resources such as agents and services is rapidly increasing on the Internet, it is ineffective for every agent to go through a centralized intermediary to form trust relationships. In addition, the degree of trust on a trusted authority may drop as the number of trustees increases. Thus, distributed trust can be used to complete current security practices and manage trust more effectively. The above two views are complementary rather than contradictory. In the light that agents are highly dispersed on the Semantic Web, it is natural to adopt the distributed model of trust for Semantic Web agents. However, as trust evolves, well-agreed trustees may emerge, which form local centroid of agent communities. Agents have their options of going for locally centralized trust models.

3. Trust based knowledge outsourcing

Trust based knowledge outsourcing involves several practical design issues: (1) how to dynamically learn trust; (2) how to use trust in search and justification; and (3) how public trust affects knowledge outsourcing.

3.1 Learning Trust

When outsourcing knowledge, it is important to estimate the trustworthiness of the participants before the initiator uses their answers. The trust estimation is indeed a social learning problem. Conte [9] suggested two types of social learning: social facilitation and imitation. *Social facilitation* occurs when an agent derives trust by reasoning about its own knowledge and past social experiences. *Imitation* occurs when an agent derives trust by mimicking other "good" agents, e.g. A trusts B because many other agents trust B.

Trust relations can be learned from *prior social knowledge*. Based on prior social knowledge and personalized decision rules, an agent could directly derive trust relations. For example, if an agent A believes that (1) agent B is a doctor and (2) that all doctors' medical advice can be trusted, then it can derive B's medical advice can be trusted even A has never met B. Prior social knowledge can be obtained from either the human user (e.g. a written profile about friendship relation expressed in the "friend of a friend"(FOAF) ontology) or the machine learning results from the Web (e.g. ReferralWeb extracted Know relation from co-occurrence of names on WWW pages [19]). In fact, this method is the essence of trust management.

One can also learn trust relations from *past social experience*. By labeling the past experience with "right", "wrong", or "unknown", an agent can dynamically learn first-hand trust by using machine learning methods, e.g. reinforcement learning [18, 27]. There are two commonly used methods: histogram and feedback-function. *Histogram* counts experience and derives trust from success rate. However, it doesn't "forget" history, i.e., it can't capture the change of agent behavior. *Feedback-function* derive trust from recent experience. We defined a "inflation function" , which increases or decreases trust values as each knowledge outsourcing event occurs. In the following formula, R(A,B,t) refers to the reliability derived by A about B at logical time t (each knowledge outsourcing event increases logic time by 1).

$$R(A,B,t) = \begin{cases} default\_trust & if\ t = 0 \\ 1 - R(A,B,t-1) * (1 - INC) & elseif\ Right(B) \\ R(A,B,t-1) * DEC1 & elseif\ Wrong(B) \\ R(A,B,t-1) * DEC2 & elseif\ Unknown(B) \end{cases}$$

Trust relations can be also derived from *the other agents' consensus*. While the above two methods belong to social facilitation, this method is an imitation. This method uses the consensus of selected "trusted" agents to derive trust about an unfamiliar agent. By assuming most agents are "trustable", an agent can use the weighted consensus of randomly selected agents to derive both domain trust and referral trust. The formula below shows that an agent A derives the reliability of domain trust about agent B $R_{domain}(A, B, domain)$ from the weighted consensus of a set of other agents NA. It is notable that reputation authorities are always good referrers.

$$\mathrm{R_{domain}}(A, B, domain) = \frac{\sum_{N \in NA} [\mathrm{R}_{domain\_ref}(A,N,domain) * \mathrm{R}_{domain}(N,B,domain)]}{|NA|}$$

3.2 Trust-based consensus query

Aiming at knowledge outsourcing, we propose a friend consensus (FC) approach. FC is expected to address the open issues in the search and justification steps of knowledge outsourcing. In the search step, only the initiator can initiate inquiries. Therefore we could avoid unintentional trust caused by delegation and have better control over search. In the justification step, the answer results from the consensus that emerges from a group of relevant friends. The consensus is more reliable and unbiased than simply fully trusting a single agent. Yu *et al.,* [31] suggested a similar method, but didn't consider inconsistent answers and trust was not taken into account.

The FC algorithm helps an agent S to find the "right" binary truth value for a query Query. S always asks a set of relevant and reliable domain experts for answers. If the experts cannot reach consensus (i.e. either due to insufficient expert population (more than T) or lack of consensus among the existing experts), it asks reliable referrers to modify the expert set (i.e. find new experts or change trust value of existing experts). The algorithm could fail when the agent can't modify the expert set and the experts can't reach consensus.

```
Friend-Consensus (S, Query, T)
1. E = SelectRelevantExperts(S, Query)
2. R= SelectRelevantReferrers(S, Query)
3. Searched= {S}, depth =0
4. while (not Consensus(S, E, T) )
5.      if (not canProceed( R, depth) ) then return fail;
6.      NewR = {}
7.      for each r in R
8.        RefE = SelectRelevantExperts(r, Query)
9.        RefR = SelectRelevantReferrers(r, Query )
10.       Experts = Merge(Experts, RefE, r.w)
11.       NewR = Merge(NewR, RefR, r.w)
12.     end for
13.     Searched = Searched U R
14.     R = NewR − Searched
15.     FilterReliable(R, S)
16.     depth++
17. end while
18. return success

Consensus (S, Expert, T, Searched)
1. RelExperts = FilterReliable( Expert, S )
2. if (|RelExperts| < T) then return false;
3. Inquire RelExperts who haven't been inquired.
4. if more than half agents have same answer
5.    then return true
6.    else return false
```

The search complexity is controlled by following heuristics: (1) Relevant heuristic. Only relevant agents (e.g., those believed to have domain knowledge relevant to the Query) are asked. This heuristic reduces search branches forked from the hub-like agents (e.g. reputation authorities). (2) Reliable heuristic. Every agent has a threshold that determines which agents it trusts. Agents do not ask unreliable agents, i.e., agents falling below this threshold. (3) Short referral distance heuristic. The *referral distance* between two agents is the length of shortest *trust path* – a sequence of referral trust. The agents with shorter referral distance are always more reliable. (4) Small world heuristic. The small world phenomenon [23, 28] shows that the shortest trust path between any two agents is nearly constant, i.e. six degree. This heuristic bounds search depth while preserving search completeness. (5) No delegation heuristic. This method guarantees that the inquiry agent has full control over outsourcing and could avoid unintentional trust.

This FC algorithm depends on the trust knowledge learned from past experiences and existing social knowledge, and can fail in absence of trust knowledge. Therefore we use some alternative algorithms that can obtain results and accumulate trust knowledge when trust knowledge is rare.

3.3 Public trust and Productive Agent society

As the size of an agent society becomes large, agents will more frequently interact with unfamiliar agents. To avoid being mislead or deceived by "bad agents", the agent needs a "correct" opinion about an unknown agent from external sources. A productive agent society should have well-known reputation authorities which can always suggest "right" public trust and avoid "wrong" ones. A reputation authority learns public trust from the second-hand social experiences reported by all agents in the society. The collaborative rating approach is simple and widely adopted by commercial web sites such as eBay [25], and Amazon. The formula below shows how the public trust about agent A as expert in domain D is derived from a group of agents, which is denoted by S. However, this measurement is

only reliable when $|S|$ is large enough, i.e., lots of social experience has accumulated.

$$\text{Reputation}_{\text{expert}}(\text{A}, \text{D}) = \frac{\sum_{P \in S} \text{R}(P, A, D)}{|S|}$$

4 Experiments

We conducted a series of controlled experiments to simulate the evolution of trust and to study how trust affects knowledge outsourcing. This section describes some of the results.

4.1 Experimental Settings

The knowledge outsourcing simulation is conducted in an agent community which consists of 50 individual agents. Each agent has a domain knowledge base consisting of truth values (true, false, or unknown) for a set of propositions and a social knowledge base recording trust knowledge. In each knowledge outsourcing activity, an agent is randomly picked as the initiator. One test round has 500 activities, and normally the output of experiment became stable after 50 rounds have passed.

An agent's domain knowledge is controlled by two properties: *Real-Coverage*, the fraction of propositions in the domain for which the agent has a committed belief, and *Real-Accuracy*, the fraction of committed beliefs that are correct. For Real-Coverage, we tried two types of distributions: *Normal Distribution* and *Zipf distribution* (see Figure 2). They are chosen because we believe they are similar to real world knowledge distribution in increasing order. For Real-Accuracy, we tried two cases: *Uniform accuracy*, which make all agents have same accuracy, and *Rich-get-richer accuracy*, which make an agent's Real-Accuracy proportional to its knowledge coverage, i.e., larger knowledge coverage cause higher accuracy. In the following context, we use the tuple of (Real-Coverage, Real-Accuracy) to generate four test cases which are denoted by initials, e.g. (Uniform, RGR) denotes the test case which use "Uniform knowledge distribution" and "rich-get-richer accuracy". Moreover, we also assume each agent has a static domain knowledge base, i.e. it does not update its domain knowledge during knowledge outsourcing.

We tested four consensus query algorithms: (1) K-Random(KR) simply randomly selects k agents and uses the uniform weighted consensus of their opinions; (2) K-Random-Learner1 (KRL1) further weights agents' opinions with learned domain trust; (3) K-Random-Learner2(KRL2) further uses the k agents' referral trust to refine the initiator's domain trust; (4)

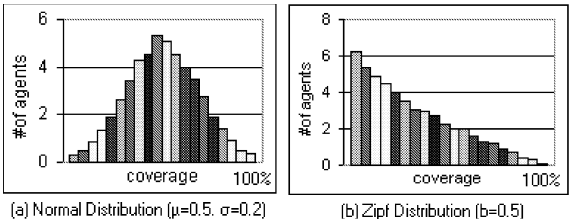

Figure 2. Two knowledge distributions

Friend Consensus(FC) uses the algorithm described in section 3.2. All these algorithms could stop running before reaching a consensus, and some algorithms require prior learned trust knowledge, so we did an additional "backup" consensus when the first consensus output "unknown". In particular, KR, KRL1, and KRL2 were backed up by themselves, and FC by KRL2.

4.2 Consensus Query

In our experiment, the consensus query algorithms are evaluated with three quantitative metrics: (1) *success rate* is the fraction of queries that have correct consensus. (2) *error rate* is the fraction of queries that have wrong consensus. (3) *search complexity* is the average number of nodes searched per query. In addition, we will discuss these metrics for both the first pass and the backup pass of consensus.

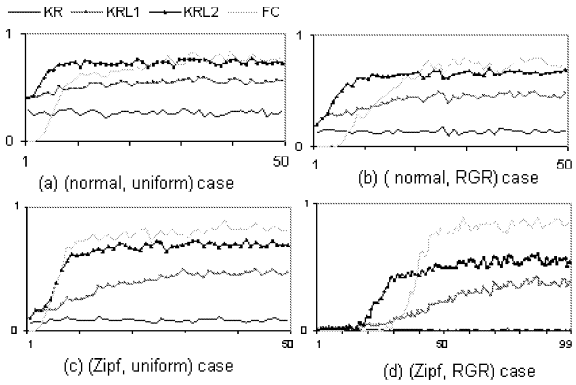

Figure 3. Success rate comparison

Success rate is measured under different knowledge distribution settings. In figure 3, the x axis shows the number of the test round, and y shows corresponding success rate. The results show that the backup consensus always has positive effect on the success rate. It also shows that trust based algorithms always yield better performance than the KR. Having observed enough samples, KRL2 and FC performs well because they

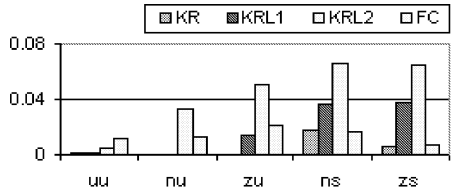

Figure 4. Error rate comparison

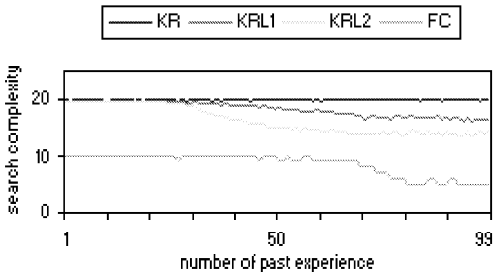

Figure 5. Search complexity for (Zipf, RGR)

tend to visit best domain experts by using learned referral trust from the other agents.

Error occurs when algorithm's assumptions fail. The random consensus algorithm uses the assumption that "agents seldom lie", and it fails in RGR knowledge distribution. The trust based algorithms have the assumption that "experts know a lot and seldom lie", and it is close to our real life experience. However, even the best expert will sometimes produce wrong answers and it is risky to propagate one voice to the whole agent society. So we use K (> 1) experts consensus in FC to reduce the error rate independent of the knowledge distribution. Figure 4 shows that the error rate of FC is lower than that of KRL2. KR has zero error rate because it inquires large enough number of agents without considering communication cost.

Search complexity is dominated by two factors: *search branching factor* and *first pass consensus failure rate*. The search branching factor is ten for KR, KRL1, and KRL2, and three for FC. We found that K is always larger than T. That is because enough samples are needed to guarantee low error rates if we choose recommenders randomly. However, the FC algorithm can consult with a limited number (perhaps a constant) of experts to get correct answers. In addition, the backup consensus introduces more search nodes proportional to the failure rate of the first pass. Figure 5 compares the search complexity in (Zipf, RGR) knowledge distribution where most agents are ignorant. The KR algorithm has the highest search complexity because of high failure rate caused by the overwhelming ignorance. The FC algorithm has lowest search complexity because it only inquires a few experts and its fail rate of the first pass consensus decreases after enough trust knowledge has been learned. The KRL2 algorithm has low search complexity because it has low failure rate.

4.3 Trust evolution within agent community

Our experiment shows that FC algorithm can improve the success rate and reduce search complexity. But we want to learn how trust knowledge evolves in the agent society. In agent community with (Normal, Uniform) knowledge distribution, we map the agent society to a direct graph, where each arc maps to trust relation between agents and each node maps to an agent. Then we found that the best domain experts always have a large number of in-links, i.e. they are pointed by many other agents.

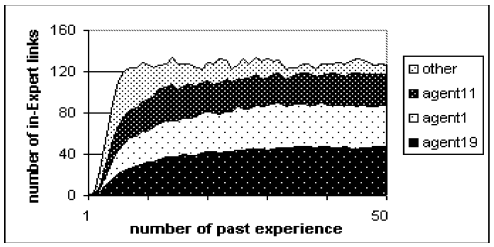

Figure 6. In-link distribution

Table 1. Top 3 experts' knowledge distribution

Rank	Coverage	Accuracy	Agent ID
1	0.92	0.967391	Agent19@foo.com
2	0.91	0.956044	Agent1@foo.com
3	0.9	0.955556	Agent11@foo.com

Figure 6 shows the change of in-link distribution during trust evolution. Table 1 shows the top three domain experts' real knowledge distribution. The result shows that (1) an agent's in-link measure is highly related to its rank, and (2) the top three agents always occupy the most in-links. Does Zipf's law distribution exist in the number of an agent's in-link and its expertise ranking? We are not yet in a position to draw such a conclusion due to the small amount of experiment data we have. Nonetheless, we believe that when trust is learned and propagated in knowledge outsourcing, the agent will be able to very quickly and easily detect the authority with FC and KRL2 algorithms.

5. Related Work

Collaborative filtering systems [5, 24] enable people to make decisions based on recommendations under the assumption that people who have similar interests will generate similar ratings. In this approach, a user uses weighted average of recommended ratings to predict his ratings, and the weight comes from the correlations between two user-interests. However, the user correlation is not the only factor of weight. Yu *et al.,* [31] presented a social network approach which used *expertise* and *sociability* to derive the weight. While expertise is used to compute user correlations in the sense of shared interests, sociability concerns about the other agent's referral capability. By considering the referral social network, their methods greatly improved the information access performance. Though their work remained at an empirical stage, the learning methods did produce the statistical premise for creating trust in agent society.

Trust in Multi-agent system focuses on formal models that capture trust in agent society [22]. Most models are based on strong social foundation, which is difficult to automate. Some recent work [1, 32] puts trust in the context of ecommerce or a virtual community, and adapts trust models to practical implementation by allowing some global information to be maintained at each individual agent or relying on a centralized server. But maintaining a global picture of the social network is too expensive or impossible for a single agent, therefore, peer-to-peer trust [2, 29] has been proposed as a mechanism for building trust in P2P electronic communities. Nonetheless, they do not address the issue of uniquely identifying peers over time and associating their history with them.

Trust in the Semantic Web is being actively studied recently. Starting from a distributed trust management [4] model, Finin and Joshi [12] suggest a semantic policy language to manage delegation on Semantic Web. Viewing the web as a social network, Golbeck et al., [14] proposed trust network on the Semantic Web and observe that the small-world phenomenon can be used to reduce the search complexity. However, their general network flow analysis approaches to trust propagation do not scale well. To facilitate expressing and inferring social relation on Semantic Web, the FOAF project suggested Resource Description Framework (RDF) based language to describe friendship information [10]. The FOAF vocabulary contains "see also" pointers to link to other FOAF files. This not only supports finding documents, service, and data on the Web based on their properties and relationships but also provides a basis for learning about new people based on their experience and interest.

Our approach may be particularly appropriate for modeling knowledge of what John Searle [26] calls "social facts". Searle is a "realist" in that he believes that there is a unique real word and facts are objective assertions about this world and can be, in principle, verified as being true or false. However, Searle identifies two categories of objective facts – "brute facts" and "social facts". Brute facts exist independent of what people believe about them, e.g., that hydrogen atoms typically have one electron. Social facts are facts whose existence depends on the mental states of humans, i.e., they are facts only when people believe them to be facts. Examples of social facts include that a particular piece of paper is a thousand dollar bill and that George W. Bush is the 43rd president of the United States. Clearly the validity of a social fact depends on its being part of a consensus model for the relevant community of agents.

6. Conclusions and Future Work

This paper discusses trust in the context of knowledge outsourcing on the Semantic Web. Instead of using the commonly accepted triplet notation, it describes a quadruplet (trust, distrust, unknown, ignorance) to better capture the trust value. Transitivity of trust is determined by the usage of trust knowledge. We also incorporate reputation in the framework of trust to complement the personal trust that is directly derived from first-hand experience. The evolution of trust illustrates how small world phenomenon occurs in the so-called trust network.

Our trust-based knowledge outsourcing approach is developed with the consideration of complexity reduction and optimization. We discuss several methods to derive personal trust from social knowledge or experience. We also suggest in our friend consensus algorithm that trust-based knowledge outsourcing should be dynamically derived from both the initiator's and other agents' personal trust. Our consensus based justification mainly relies on the consensus aspect of justification. Other factors of justification such as logical reasoning based belief justification may also be considered in future.

Some preliminary experiments were conducted to validate the approach for knowledge outsourcing. The results confirmed that agents can get better information and get it more efficiently by using more trust knowledge. We also found that the trust-based friend consensus algorithm reduces the error rate despite the potential for propagating of incorrect information. Future experiments will be carried out to further explore

the role of trust in a large-scaled and multi-domain multi-agent society, where we believe the reputation authorities will emerge as a result of trust evolution.

References

[1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *HICSS*, 2000.

[2] K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In *CIKM*, pages 310–317, 2001.

[3] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[4] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 164–173, 1996.

[5] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[6] C. Castelfranchi and Y.-H. Tan. *Trust and Deception in Virtual Societies*. Kluwer Academic Publishers, 2001.

[7] B. Christianson and W. S. Harbison. Why isn't trust transitive? In *Proceedings of the Security Protocols Workshop*, pages 171–176, 1996.

[8] Y.-H. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick, and M. Strauss. Referee: Trust management for web applications. *World Wide Web Journal*, 2(3):127–139, 1997.

[9] R. Conte and M. Paolucci. Intelligent social learning. *Journal of Artificial Societies and Social Simulation*, 4(1), 2001.

[10] E. Dumbill. Finding friends with xml and rdf, 2002.

[11] C. English, P. Nixon, S. Terzis, A. Mcgettrick, and H.Lowe. Dynamic trust models for ubiquitous computing environments. In *Workshop on Security in Ubiquitous Computing*, 2002.

[12] T. Finin and A. Joshi. Agents, trust, and information access on the semantic web. *ACM SIGMOD Record*, 31(4):30–35, 2002.

[13] D. Gambetta. *Trust: Making and Breaking Cooperative Relations*. Department of Sociology, University of Oxford, 2000.

[14] J. Golbeck, B. Parsia, and J. Hendler. Trust networks on the semantic web, 2003.

[15] T. Grandison and M. Sloman. A survey of trust in internet application. *IEEE Communications Surveys & Tutorials (Fourth Quarter)*, 3(4), 2000.

[16] J. Hendler. Agents and the semantic web. *IEEE Intelligent Systems*, 16(2), 2001.

[17] A. Josang. An algebra for assessing trust in certification chains. In *Proceedings of the Network and Distributed Systems Security (NDSS'99) Symposium, The Internet Society*, 1999.

[18] L. Kaelbling, M. Littmna, and A. Moore. Reinforcement learning a survey. *Journal of AI Research*, 4:247–285, 1996.

[19] H. A. Kautz, B. Selman, and M. A. Shah. The hidden web. *AI Magazine*, 18(2):27–36, 1997.

[20] R. A. Malaga. Web-based reputation management systems: Problems and suggested solutions. *Electronic Commerce Research*, 1(4):403–417, 2001.

[21] S. Marsh. Trust and reliance in multi-agent systems: A preliminary report. In *Proceedings of 4th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pages 94–112, 1992.

[22] S. P. Marsh. *Formalising trust as a computational Concept*. Ph.D. dissertation, University of Stirling, 1994.

[23] S. Milgram. The small world problem. *Psychology Today*, 1(1):60–67, 1967.

[24] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM Press, 1994.

[25] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[26] J. R. Searle. *The Construction of Social Reality*. Free Press, 1995.

[27] T. Y. Tang, P. Winoto, and X. Niu. Who can i trust? investigating trust between users and agents in a multi-agent portfolio management system. In *Proceedings of the AAAI-02 Workshop on Autonomy, Delegation, and Control: From Inter-agent to Groups*, 2002.

[28] D. Watts. *Small worlds : the dynamics of networks between order and randomness*. Princeton University Press, Princeton, 1999.

[29] L. Xiong and L. Liu. Building trust in decentralized peer-to-peer electronic communities. In *Proceedings of the Fifth International Conference on Electronic Commerce Research*, 2002.

[30] B. Yu and M. P. Singh. Distributed reputation management for electronic commerce. *Computational Intelligence*, 18(4):535–549, 2002.

[31] B. Yu, M. Venkatraman, and M. P. Singh. An adaptive social network for information access: Theoretical and experimental results. *Journal of the Applied Artificial Intelligence*, 17(1):21–38, 2003.

[32] G. Zacharia, A. Moukas, and P. Maes. Collaborative reputation mechanisms in electronic marketplaces. In *HICSS*, 1999.