Building the Semantic Web by Mass Collaboration

Matthew Richardson†
University of Washington
Box 352350
Seattle, WA 98195-2350
mattr@cs.washington.edu

Rakesh Agrawal
IBM Almaden Research Center
650 Harry Road
San Jose, CA
ragrawal@almaden.ibm.com

Pedro Domingos
University of Washington
Box 352350
Seattle, WA 98195-2350
pedrod@cs.washington.edu

ABSTRACT

Though research on the Semantic Web has progressed at a steady pace, its promise is yet to be realized. One major difficulty has been that the Semantic Web requires global consistency and quality, properties that are nearly impossible to achieve in such a large scale, mass collaborative system. We circumvent these problems by adopting instead the notion of local consistency. We propose individual maintenance of beliefs and trusts in other users. This paper defines global properties for combination functions which merge such trusts and beliefs, and defines a class of functions for which merging may be done locally while being equal to having done the merging locally. We show an equivalence between merging trusts and merging beliefs, allowing use of whichever is more efficient. We give examples of specific functions which fit into the framework and apply them to data from the Epinions knowledge-sharing site and our implemented, real-world testbed, the BibServ bibliography server. Experimental results confirm that the methods are robust to maliciousness and noise, and do not require unreasonably high expectations of user quality. We hope that these methods will help move the Semantic Web closer to fulfilling its promise.

Keywords

Semantic Web, trust, belief, mass collaboration, web of trust

1. INTRODUCTION

Since the articulation of the vision for the Semantic Web in the landmark paper [10], the Semantic Web has become the focus of research on building the next web. The philosophy behind the Semantic Web is the same as that behind the World-Wide web – anyone can be an information producer or consume anyone else's information. The key difference between the Semantic Web and the WWW is that in the Semantic Web, this content is intended to be machine understandable.

Thus far, most research for the Semantic Web [34][6] has focused on defining standards for communicating facts, rules, ontologies, etc. XML, RDF, RDF-schema, and others form a necessary basis for the construction of the Semantic web, but none of them address the major issue of how can we ensure the quality of the information being provided? As with the World-Wide Web, we expect there to be problems of information quality, consistency, redundancy, and relevance on the Semantic Web. Consider the following scenarios:

- **Incorrect Information**: Suppose you wish to find the elevation of Mt. Everest. One source may claim 29,035 feet, while another may claim 29,028 feet[1]. Which source is more trustworthy? Whether intentional or not, misinformation is likely to be prevalent.
- **Disagreement**: Suppose you wish to learn about gravity waves. This is an area of active research, in which there is disagreement, even between highly credible scientists.
- **Difference of Opinion**: Suppose you wish to ask which graduate school has the best Computer Science department. This is a very opinionated query, and the best answer may depend on who is asking the question (for instance, what do they value? Who's opinion do they trust, etc).
- **Too much information**: Suppose you wish to find a good book on taking care of cats. A list of all books on taking care of cats is simply too much. What you really desire is a list of book that are recommended, perhaps by other users who share similar beliefs and opinions to yourself.

On the World-Wide Web, these problems have been addressed by search engines, moderated taxonomies, etc., but the problems are likely to prove much more troublesome on the Semantic Web. If it becomes as widespread as the WWW, we expect that there will be significantly more facts on the Semantic Web than there are currently pages on the WWW. Further, computers are typically more sensitive to mistakes than humans. Also, research has shown that users depend strongly on the visual appearance of web pages to estimate their credibility [19], but this is impossible on the Semantic Web.

One solution would be to require all information contained on the Semantic Web to be consistent, organized, and of high quality. But due to its sheer magnitude and diversity of sources, this will be nearly impossible. Much as in the development of the WWW, in which there was no attempt made to create a globally consistent library of information, we argue that it is neither feasible nor desirable to maintain globally consistent information on every possible topic.

Imagine what it would mean to require global consistency on the World Wide Web. It would mean, for example, that before creating a link to the best modern painting from her page, Jane first get a universal agreement on what a modern painting is and what their universal rankings are. From Jane's point of view, a painting is modern because Jane says so, and it is the best modern painting,

WWW 2003, May 20-24, 2003, Budapest, Hungary.
ACM xxx.

† Researched while at IBM Almaden Research Center.

[1] The height of Mt. Everest was widely held to be 29,028 feet until 1999, when new, more accurate GPS equipment revealed the actual height to be 29,035 feet.

again because she thinks so. The foundation of the success of the original web was the respect for individual opinions and judgments, and we do not want to lose sight of this simple axiom.

It seems clear that we will simply have to accept the existence of inconsistency, and develop methods that work under the assumption that the information will be of varying quality. On the WWW, researchers have found that one way to handle this problem is to take advantage of the link structure between pages, which can be taken to mean that the source page considers the destination page to be of good quality. Only by taking advantage of these completely distributed, mass collaborative "statements" of quality can we ever expect the entire web to be ranked.

We propose that the same approach will work on the Semantic Web, for maintaining not just quality, but also consistency, relevance, redundancy, and vocabulary differences. We argue that what is needed is only local consistency, meaning that you can understand and translate the information of a small number of "neighbors" – users who themselves also communicate with and understand a small number of people, and so on recursively such that you have access to potentially everything on the Semantic Web.

The recent popularity of peer-to-peer networks dovetails nicely with what we are proposing. Here, information "on the Semantic Web" comes to you filtered through your explicitly chosen neighbors. In this way, users only need to ensure that there is consistency between themselves and their neighbors. We view this as akin to the Piazza project [23] which forms apparent global consistency by transitive local consistency between peers. All of trust decisions are local. Jane simply specifies who she trusts and how much, as do John and Mary and others. Similarly, Jane simply creates her information without worrying about who might want to use it. Jane's trusts connect her to a world-wide network of trust, giving her access to a knowledge base which is a superset of what she herself created. The pieces she gets from others are determined by who she trusts and who her trustees trust in turn. Note that centralized information repositories on specialized topics (e.g. Yahoo, CiteSeer, dictionaries, etc.) can be seamlessly integrated in the proposed approach.

In Section 2 we formulate a model which explicitly has the dual notions of *trust* and *belief*. Trust is computed transitively so that users of the system may calculate beliefs for any statement reachable through paths of trust relations (referred to as combined beliefs). Then, in Section 3, we describe on a global level what is the meaning of belief combination for a broad class of functions. We then show that beliefs may in fact be combined on a local level while still maintaining their global interpretation. We also show a correspondence between combining beliefs and combining trusts which conveniently allows for selection of whichever is more efficient for the given system. In Section 4, we present an alternative interpretation based on probability. As with Section 3, we then proceed to show that it is possible to combine beliefs locally while maintaining this global interpretation and show a correspondence between combined beliefs and combined trusts. In Section 5 we show that the two interpretations are closely related.

Before concluding with related and future work, we give the results from two sets of experiments. In Section 6.1 we give the results from two sets of experiments. The first set uses the Epinions web of trust; its results show that belief merging works across a wide variety of user quality, levels of maliciousness and amounts of noise in estimating trust values. In Section 6.2 we present the results of experiments with our ongoing project, BibServ. BibServ is a publicly-accessible bibliography server which uses our belief and trust merging techniques to maintain quality and consistency of bibliographies. Experimental results indicate positive trends in the use of beliefs for determining the quality and relevancy of bibliography entries.

2. MODEL

In this section we introduce the general terminology and model which will be used throughout this paper.

2.1 Relation to the Semantic Web

The vision of the Semantic Web encompasses all forms of knowledge and information. For instance, content could be in the form of logical rules, factual assertions, ontologies, probability distributions, etc. All of these forms are useful for one type of task or another, so we do not wish our to discriminate between them. We thus refer to information with the generic term *statement*. A *statement* could be a single rule, or an entire ontology.

Conceptually, queries on the Semantic Web are answered by retrieving all relevant statements, and using the appropriate inference methods to bring them together and produce an answer. However, typically not only would an answer be desired, but also some form of confidence in the answer. We envision that the result of the inference is a statement with an attached *belief*, which is some function of the beliefs of the statements that were input into the inferring machinery. The way in which the inference combines beliefs is specific to the form of information it is processing, and we do not address this issue here. Instead, we address the issue of how to discount and combine the input beliefs, which may have come from multiple different users. In order to do this combination, we introduce the notion of *trust*, and the network of trusts which form a *web of trust*.

Example: Alice wants to know whether eggs are needed to make chow mein. Both Bob and Charley tell her that eggs are needed to make noodles, and Dawn tells her that there are possibly noodles in chow mein. How much should Alice trust Bob and how much should she trust Charley? How much should she believe that eggs are needed considering both Bob and Charley stated it was so? How uncertain should Alice be that there are noodles in chow mein considering Dawn was uncertain? These are all questions that this paper addresses. This paper does not answer the question of how the beliefs of the input rules should be combined in order to determine a belief for result, which is already an active area of research for many different forms of knowledge (see Section 7, Related Work).

2.2 Users, trusts, and beliefs

Consider a system of N users $U=\{u_1, u_2, \ldots, u_N\}$, who, as a whole, have made M *statements*. A statement is any assertion made by some user, whether accurate or not. Statements may be factual (*The sky is blue*), controversial (*Black holes do not exist*), opinionated (*Chocolate is the best flavor of ice cream*), ontological (*Mammal is a subclass of Animal*), translations between terms (*When John says 'azul', he really means 'blue'*), etc. We assume for convenience that statements are globally enumerated, though this is not necessary in practice. Since statements are considered independently, without loss of generality we introduce the system as if there is only one statement.

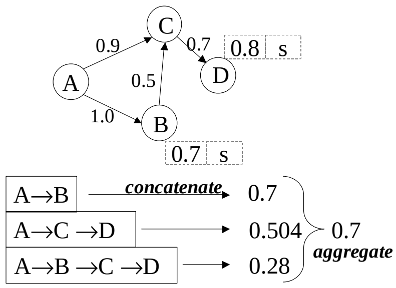

Figure 1: An example web of trust, demonstrating path algebra belief merging. Edges are labeled with trust. 0.7 and 0.8 represent B and D's personal beliefs. A's belief is the aggregation of the concatenation of trusts and beliefs for any path from A to any node with belief. Concatenation and aggregation are multiplication and addition.

2.2.1 Beliefs

Any user may assert her *personal belief* in the statement. We will assume that a belief is a value taken from [0,1], where a high value means that the statement is accurate, credible, and/or relevant. Beliefs will sometimes be interpreted as a probability. Let $\mathbf{b}_i$ represent user i's personal belief. If i has not provided one, we set $\mathbf{b}_i$ to 0. We will often refer to the collection of personal beliefs as the column vector $\mathbf{b}$

2.2.2 Trusts

A user i may specify a *personal trust*, $\mathbf{t}_{ij}$, for any user j. We will assume that trust is also a value taken from [0,1], where a high value means that the user is credible, trustworthy, and/or shares similar interests and beliefs. Take $\mathbf{t}_{ij}$ to be 0 if i does not provide a personal trust value for j. Note that $\mathbf{t}_{ij}$ need not equal $\mathbf{t}_{ji}$ The collection of personal trusts of all users can be represented as a N×N matrix $\mathbf{t}$. We write $\mathbf{t}_i$ to represent to row vector of user i's personal trusts in other users.

2.2.3 Merging

The purpose of the web of trust is to provide a structure on which we may compute, for any user, their belief in the statement. We will refer these beliefs as *merged beliefs* (**B**), to distinguish them from the user-specified *personal beliefs* (**b**). We also define the trust between any two users to be the users' *merged trusts* (**T**), as opposed to the user specified *personal trusts* (**t**) which define their set of neighbors. Note that although it is more typical for the case of a letter to denote whether it is a vector or a matrix, we have chosen to let case denote whether the values are personal or merged, and remind the reader to keep in mind that the letter 'b' always denotes vectors and the letter 't' always denote matrices.

3. PATH ALGEBRA INTERPRETATION

In this section, we define a global interpretation for merged beliefs based on the paths between users and beliefs. In Section 4 we consider an alternative probabilistic interpretation. In both sections we will show that merged beliefs may be computed using only local information, yet still hold true to their global definition. We will also show that in both cases, merged beliefs may be computed indirectly by first computing merged trusts, and then applying them to users' personal beliefs.

3.1 Global belief merging

We first define the desired global meaning of belief combinationin the network of trusted users. For the moment, we consider only acyclic graphs (we later discuss application to cyclic graphs). Refer to Figure 1. User A wishes to determine her belief in some statement, which may be believed by multiple users (B, D in this example). In this interpretation, we specify that the merged belief is some function of the collection of all paths between the user and every user which believes the statement. Borrowing terminology from the generalized transitive closure literature on path computations [3], the specific computation is as follows:

1. Enumerate all (possibly exponential number of) paths between the user and every user with a personal belief in the statement.
2. Calculate the belief associated with each path by applying a *concatenation function* to the trust values along the path and also the personal belief held by the final node.
3. Combine those beliefs with an *aggregation function*

We will see shortly that as long as the concatenation and aggregation functions have certain properties, the potentially exponential number of paths between the user and beliefs does not cause the computation to become intractable.

What happens if the user herself has some belief in the statement? As we will see later, the computation can be applied to graphs with cycles. By specifying a trust in herself, a user's personal beliefs may be incorporated into her merged beliefs (see Section 3.5 for a more thorough discussion).

Some useful concatenation functions are multiplication and minimum value. Some possible aggregations functions are addition or maximum value. Various combinations lead to plausible belief-merging calculations such as considering the most-reliable path or maximum flow between the node and the statement.

For brevity, we will let ○ and ◊ represent the concatenation and aggregation functions respectively (for intuition, think 'multiplication' and 'addition'). For example, since $\mathbf{t}_{ik}$ is the amount that user i trusts user k, and $\mathbf{t}_{kj}$ is the amount that k trusts j, then $\mathbf{t}_{ik}\circ\mathbf{t}_{kj}$ is the amount that user i trusts user j *via* k. The aggregation function, ◊, is used to combine multiple paths. For example, the amount that i trusts j via any single other node would be $\Diamond(\ \forall k\text{: } \mathbf{t}_{ik}\circ\mathbf{t}_{kj}\ )$. If ◊ is addition and ○ is multiplication, we will have $\Diamond(\ \forall k\text{: } \mathbf{t}_{ik}\circ\mathbf{t}_{kj}\ ) \equiv \sum_k \mathbf{t}_{ik}\mathbf{t}_{kj}$. We define the matrix operation $\mathbf{C}=\mathbf{A}\bullet\mathbf{B}$ such that $\mathbf{C}_{ij} = \Diamond(\ \forall k\text{: } \mathbf{t}_{ik}\circ\mathbf{t}_{kj}\ )$. Note that for the previous example, $\mathbf{A}\bullet\mathbf{B}$ is simply matrix multiplication.

3.2 Local belief merging

The global meaning of trusts given above assume A has full knowledge of the network including the personal trusts between all users. Not only does this raise major privacy concerns, but it also runs counter to the ideas of local consistency and keeping the network of trust distributed. Can we instead merge beliefs locally while keeping the same global interpretation?

Following [3], let *well-formed decomposable path problems* be defined as those for which ◊ is commutative and associative, and ○ is associative and distributes over ◊. All of the examples for aggregation and concatenation given above result in well-formed path problems. Any such well-formed path problems may be computed using *generalized transitive closure* algorithms[2], which use only local information. One such algorithm infers beliefs by starting with the personal beliefs and iteratively incorporating the neighbors' beliefs until the network converges. If we let **B** represent the vector of merged beliefs, then the algorithm is as follows:

1. $\mathbf{B}^{(0)} = \mathbf{b}$
2. $\mathbf{B}^{(n)} = \mathbf{t}\bullet\mathbf{B}^{(n-1)}$, or alternatively, $\mathbf{B}_i^{(n)}=\Diamond(\ \forall k\text{: } \mathbf{t}_{ik}\circ\mathbf{B}_k^{(n-1)}\ )$
3. Repeat until $\mathbf{B}^{(n)} = \mathbf{B}^{(n-1)}$

(where $\mathbf{B}^{(i)}$ represents the value of **B** in iteration *i*.)

Notice that in step 2, the user needs only the merged beliefs of her immediate neighbors. This computation allows us to merge beliefs *locally* while keeping the same *global* interpretation. What happens if we choose concatenation and aggregation functions that do not form a well-formed decomposable path problem yet still apply the algorithm above? In this case, the global meaning of beliefs described in Section 3.1 no longer holds and unexpected behavior may result.

Refer to Figure 2 (*Case I*). Imagine selecting any node (with no personal belief in the statement) and removing it from the network, adding edges from any node which trusted it to any node it trusted (set the label of such an edge to be the concatenation of the trusts of the two edges it was formed from). Using the method described in Section 3.1, the merged beliefs of the remaining users will remain unchanged. We call this *weak global invariance*. We feel this is an important property as otherwise the introduction of users who do nothing more than pass through information can affect the trust between users.

We will sometimes use the term *belief combination function* to refer to the above algorithm and some selection of ○ and ◊ for merging beliefs.

3.3 Strong invariance

We can imagine another property which may be desirable. Again refer to Figure 2 (*Case II*). A trusts C through B. If we add an arc of trust directly from A to C, and the trust between A and C is unchanged, we say that the belief combination function has *strong global invariance*, to differentiate it from the *weak global invariance* introduced above.

If John computes his indirect trust in Jane, and then adds Jane as a neighbor with that same value of trust, then it is reasonable to expect his trust in Jane to remain unchanged. Arguably, in some instances, combination functions without strong global invariance make sense. For instance, if ◊ is addition, then adding a direct arc of trust to a user who is already indirectly trusted will increase the trust in that user, which is not entirely unreasonable. We believe that weak global invariance is a minimum requirement for a reasonable belief function, while strong global invariance is optional.

Any belief combination function with weak invariance, for which the aggregation function is also *idempotent* (meaning, $\Diamond(x, x) = \Diamond(x)$), will have strong invariance. This follows trivially from the fact that the aggregation function is associative. Interestingly, whether or not the aggregation function must be idempotent is the primary difference between Agrawal's well-formed decomposable path problems [3] and Carre's path algebra [12] (also related is the definition of a closed semiring in [4]). One example of a belief combination function with strong global invariance is the one defined with ◊ as maximum and ○ as multiplication.

[2] One such algorithm is the semi-naïve algorithm [7]. See Section 3.7 for a discussion on computational efficiency.

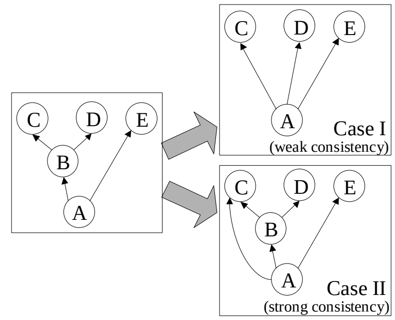

Figure 2: Strong and weak invariance.

3.4 Merging trusts

Notice that the majority of the belief merging calculation involves the concatenation of potentially long chains of trust. The belief only enters the computation at the endpoint of each path. Can we, instead of merging beliefs, consider merging trusts and then reusing these merged trusts when calculating the beliefs for different statements?

Interpretation of merged trusts

We define the interpretation of globally merged trusts in the same was as was done for beliefs: the trust between user *i* and user *j* is an aggregation function applied to the trust of every path between them, where the trust of a path is found by concatenating the trust values along it.

Local trust merging

As with merging beliefs, it falls directly from path algebra that, if ◊ is commutative and associative, and ○ is associative and distributes over ◊, then we can combine trusts locally while still maintaining global meaning:

1. $\mathbf{T}^{(0)} = \mathbf{t}$
2. $\mathbf{T}^{(n)} = \mathbf{t}\bullet\mathbf{T}^{(n-1)}$
3. Repeat until $\mathbf{T}^{(n)} = \mathbf{T}^{(n-1)}$

(We define **T** to be the NxN matrix of merged trusts, with $\mathbf{T}^{(i)}$ is the value of **T** in iteration *i*) Step 2 requires merged trusts only from immediate neighbors. The result of the computation is a set of merged trusts that maintain the global interpretation.

This leads us to the following theorem:

Theorem 1
If ◊ is commutative and associative, and ○ is associative and distributes over ◊, and **t**, **T**, **b**, and **B** are as above, then $\mathbf{t}\bullet\mathbf{B} = \mathbf{T}\bullet\mathbf{b}$ The proof is in the appendix.

Thus, for a wide class of aggregation and concatenation functions, merging trusts accomplishes the same as merging beliefs.

3.5 Cycles

Up to now, we have assumed the graph is acyclic. The problem with cyclic graphs is that there may now be an infinite number of paths between two nodes. This means the aggregation function may potentially have to be applied to an infinite number of values.

However, it is improbable that arbitrary webs of trust in the Semantic Web will be acyclic. Indeed, the Epinions web of trust (see Section 6.1) is highly connected and far from acyclic.

Again, borrowing terminology from path algebra, define a path problem as *cycle-indifferent* if it is not affected by introducing a cycle in the path between two users. As a result of cycle indifference, the aggregation over infinite paths will converge, since only the (finite number of) paths without cycles affect its calculation. By direct application of this property:

Proposition 1
All of the results, definitions, and theorems introduced thus far are applicable to cyclic graphs if ◊ and ○ define a cycle-indifferent path problem.

One example of a cycle-indifferent combination function is the one defined by using multiplication for concatenation, and maximum for aggregation. Since the beliefs and trusts all range in [0,1], concatenation of additional edges will only decrease the path's trust value. Maximum will consider only the paths which do not encounter the same node twice, as it is guaranteed to have a lower concatenated value than the same path without the cycle.

We now see how the global interpretation incorporates personal beliefs. By adding an arc of trust to herself, Jane may specify how much she trusts her own beliefs. Because the combination function must be cycle-indifferent, we know that only the "first time around the cycle" will count, namely her beliefs will be aggregated into the set of beliefs exactly once. In the case of maximum, we can see that this means that if she believes the statement more than her neighbors do (scaled by trusts), her merged belief will be based on her personal belief. Otherwise, it will be based on the neighbors' beliefs.

On cyclic graphs, a combination function which is not cycle-indifferent has the questionable property that, by modifying his own personal trusts, a user will sometimes be able to affect others' trusts in him. Also, in order to guarantee that the computation converges, one must show that the aggregation over infinite paths will converge. However, requiring that all combination functions are cycle-indifferent may be overly restrictive. In section 4 we explore an alternative interpretation for belief combination which allows the use of combination functions which are not cycle-indifferent.

3.6 Selection of belief combination function

The selection of belief combination function may depend on the application domain, desired belief and cycle semantics, and the expected typical social behavior in that domain. The ideal combination function may be user-dependent. For the remainder of the paper, we will always use multiplication for concatenation, though in the future we would like to explore other functions (such as taking the minimum value along a given path). Following is a brief summary of four different aggregation functions we have considered.

Maximum value

This aggregation function returns the maximum value of its parameters. Since this means that the trust between any two users is simply the most-trusted path between them, we consider this an *optimistic* belief combination function. The advantages of maximum are that it is cycle-indifferent and results in absolute valued beliefs. Using maximum to combine beliefs is consistent with fuzzy logic, in which maximum has been shown to be the most reasonable function for performing a generalized *or* operation over [0,1] valued beliefs [9]. The interpretation is that the user will believe anything stated by at least one of the users he trusts.

Minimum value

Similar to maximum value, minimum value is also a possible aggregation function. However, unlike maximum, minimum is not cycle-indifferent. In fuzzy logic, minimum value is used to perform the *and* operation. The interpretation is that the user will only believe a statement if it is believed by all of the users she trusts.

Average

Average does not satisfy the requirements for a well-formed path algebra outlined above (average is not associative). However, average can still be computed by using two aggregation functions: sum and count (count simply returns the number of paths by summing 1's). By passing along these two values, each node can locally compute averages. Average is not cycle-indifferent.

As a combination function *maximum* naturally handles missing values (by letting them be 0), is strongly consistent, and is cycle-indifferent, making it a good candidate for belief combination.

3.7 Computation

Since cycle-indifferent, weakly consistent combination functions are well-formed path problems, **B** and **T** may be computed using standard transitive closure algorithms. The simplest of these is the semi-naïve algorithm [7], which runs in $O(N^4)$ time, and essentially prescribes repeated application of the belief update equation. If running as a peer-to-peer system, the semi-naïve algorithm may be easily parallelized. requiring $O(N^3)$ computations per node [2]. Another algorithm is the Warshall algorithm [40], which computes the transitive closure in $O(N^3)$. Some work on parallel versions of the Warshall algorithm has been done in [2]. There has also been much research on optimizing transitive closure algorithms, such as that in databases for when the graph does not fit into memory [3]. Even though the worst case time complexity is $O(N^3)$, in practice most users will have only a very small fraction of the nodes as neighbors, and the number of iterations required to fully propagate information is much less than N, reducing the time complexity to nearly linear.

Theorem 1 allows us to choose whether we wish to merge trusts or merge beliefs. The most efficient method depends on, among other things, whether the system is implemented as a peer-to-peer network or as a central server. Other factors that determine the most efficient algorithm are the number of neighbors for a given user, the number of users, the number of statements in the system, and the number of queries made by each user.

4. PROBABILISTIC INTERPRETATION

In this formulation, we consider a probabilistic interpretation of global belief combination. The treatment is motivated by random walks on a Markov chain, which have been found to be of practical use in discovering high-quality web pages [33]. In what follows, we assume the set of trusts for a given user have been normalized.

4.1 Global belief and trust merging

We now describe the meaning of merged beliefs and trusts for user *i*. Imagine a random knowledge-surfer hopping from user to user in search of beliefs. At each step, the surfer probabilistically selects a neighbor to jump to according to the current user's distribution of trusts. Then, with probability equal to the current user's belief, it says "yes, I believe in the statement". Otherwise, it says "no". Further, when choosing which user to jump to, the random surfer will, with probability $\lambda_i \in [0,1]$, ignore the trusts and instead jump directly back to user *i*.

We define a combination method to have a *global probabilistic interpretation* if it satisfies the following two conditions:

1) $\mathbf{T}_{ij}$ is equal to the probability that, at any given step, user *i*'s random knowledge-surfer is at user *j*.
2) $\mathbf{B}_i$ is equal to the probability that, at any given step, user *i*'s random knowledge-surfer says "yes".

Intuitively, user *i* will have a high merged trust for user *j* if *j* is well-trusted by many users that are (merged-)trusted by user *i*. User *i* himself will be trusted at least λ_i (more if there are cycles of trust which point back to him). Also, user *i* will have a high belief in the statement if it is believed highly by users for which his merged trust is high. The parameter λ_i can be viewed as a *self-trust*, and specifies the relative weight a user gives to his own beliefs and trusts vs. those of his neighbors. The convergence properties of such random walks have been well-studied; the beliefs and trusts will converge as long as the network of trusts is irreducible and aperiodic [31].

The behavior of the random knowledge-surfer above is very similar to that of the intelligent surfer presented in [39], which is a generalization of PageRank that allows non-uniform transitions between web pages. What makes the calculation personalized to user *i* is the random restart (the occasional jump back to user *i*), which "grounds" the surfer to *i*'s trusts. The resulting set of trusts may be drastically different from those that would be obtained using standard PageRank since the number of neighbors will typically be small. A similar technique was used for making PageRank topic-sensitive in [24], though it uses a uniform distribution in determining which link to follow.

4.2 Computation

User *i*'s trust in user *j* is the probability that his random surfer is on a user *k*, times the probability that the surfer would transition to user *j*, summed over all *k*. Taking λ_i into account as well, we can write

$$\mathbf{T}_{ij} = \lambda_i \delta(i-j) + (1-\lambda_i)\sum_k \mathbf{T}_{ik}\mathbf{t}_{kj}$$

where $\delta(0)=1$ and $\delta(x\neq 0)=0$ and $\mathbf{t}$ is normalized so that each row sums to 1 (in this way, ignoring the random restart, $\mathbf{t}_{ij}$ is the probability that the random surfer, when currently at user *i*, will transition to user *j*). This may also be written as:

$$\mathbf{T}_i = \lambda_i \mathbf{I}_i + (1-\lambda_i)\mathbf{T}_i\mathbf{t} \qquad (1)$$

where $\mathbf{I}_i$ is the i^{th} row of the identify matrix.

In order to satisfy the global probabilistic interpretation, we need $\mathbf{B}_i$ to equal the probability that user *i*'s random surfer says "yes". This would be equal to the probability that it is on a given user times that user's belief in the statement

$$\mathbf{B}_i = \sum_k \mathbf{T}_{ik}\mathbf{b}_k \text{ , or, } \mathbf{B}_i = \mathbf{T}_i\mathbf{b} \qquad (2)$$

4.3 Local belief and trust merging

As in section 3.2, we wish to perform this as a local computation. We will show that this is possible if we consider the special case where $\lambda_i=\lambda$ is constant. From Equation 1,

$$\mathbf{T} = \lambda\mathbf{I} + (1-\lambda)\mathbf{T}\mathbf{t} \qquad (3)$$

Unrolling the recursion, we find

$$\mathbf{T} = \lambda\left[\sum_{m=0}^{\infty}(1-\lambda)^m \mathbf{t}^m\right] \qquad (\text{note } \mathbf{t}^0=\mathbf{I}) \qquad (4)$$

From Equation 2, we need **B = Tb**. Substituting Equation 4,

$$\mathbf{B} = \lambda\left[\sum_{m=0}^{\infty}(1-\lambda)^m \mathbf{t}^m\right]\mathbf{b} \qquad (5)$$

Equation 5 is satisfied by the recursive definition:

$$\mathbf{B} = \lambda\mathbf{b} + (1-\lambda)\mathbf{t}\mathbf{B} \qquad (6)$$

Thus we find that in order to compute his merged belief, each user needs only to know his personal belief, and the merged beliefs of his neighbors. Besides having intuitive appeal, it has a probabilistic interpretation as well: user *i* selects a neighbor probabilistically according to his distribution of trust, $\mathbf{t}_i$, and then, with probability $(1-\lambda)$, accepts that neighbor's (merged) belief, and with probability λ accepts his own belief. Further, we note that Equation 4 is also equivalent to the following alternate definition of **T**:

$$\mathbf{T} = \lambda\mathbf{I} + (1-\lambda)\mathbf{t}\mathbf{T} \qquad (7)$$

by this equation, we find that a user may compute his merged trusts knowing only the merged trusts of his neighbors.

The probabilistic interpretation for belief combination is essentially taking the weighted average of the neighbors' beliefs. We will thus refer to this belief combination as *weighted average* for the remainder of the paper. Note that for weighted average to make sense, the computation must specify a belief for every user. If the user has not made this belief available, then we need to impute the value. Techniques such as those used in collaborative filtering [37] and Bayesian networks [15] for dealing with missing values may be applicable. If only relative rankings of beliefs are necessary, then it may be sufficient to consider all unstated beliefs to have value 0.

5. SIMILARITY OF INTERPRETATIONS

There are clearly many similarities between the probabilistic interpretation of beliefs and the path algebra interpretation defined in Section 3. In both, the beliefs may be merged by querying neighbors for their beliefs, multiplying (or concatenating) those beliefs by the trust in each neighbor, and adding (or aggregating) them together (and including the personal beliefs in the sum for the probabilistic interpretation). The correspondence is similar for trusts. Both interpretations also allow the computation of merged beliefs by concatenating merged trusts with personal beliefs.

In fact, if we let the aggregation function be addition, and the concatenation function be multiplication, then the similarity is even more clear. The only difference between the two interpretations is due to the factor, λ. If $\lambda=0$, then Equation 6 for computing **B** is functionally the same as the algorithm for computing **B** in the path algebra interpretation. However, consider this: If λ is 0 then Equation 1 for computing $\mathbf{T}_i$ simply finds the primary eigenvector of the matrix **t**. Since there is only one primary eigenvector, this means that $\mathbf{T}_i$ would be the same for all users (assuming the graph is aperiodic and irreducible). How do we reconcile this with the path algebra interpretation?

The path algebra combination function defined by using multiplication for concatenation and addition for aggregation is not cycle independent. As a result, if we were to combine beliefs as in the path algebra interpretation, the user's personal beliefs will get "washed out" by the infinite aggregation of other users' beliefs.

Hence, as in the probabilistic interpretation, all users would end up with the same merged beliefs.

Both methods share similar tradeoffs with regards to architectural design. The methods presented in this paper are may easily be employed in a peer to peer system, though they can also be used in a central server. As peer-to-peer, we expect the system to be robust to maliciousness because a user who is not merging beliefs well, or is injecting false information into the network will be trusted less over time. Further, since the default trust in a user is 0, it is not useful for a user to create multiple pseudonyms, and users are motivated to maintain quality of information.

The web of trust calculation is not susceptible to 'link-spamming', a phenomenon in PageRank whereby a person may increase others' trust in him by generating hundreds of virtual personas which all link to him. In PageRank, the uniformly random jump of the surfer means that each virtual persona is bestowed some small amount of PageRank, which they 'distribute' to the spammer, thus increasing his rank. With a web of trust, this cheating technique gains nothing unless the user is able to convince others to trust him or his virtual personas, which we expect will only occur if they provide useful information.

6. EXPERIMENTS

In this section, we measure some properties of belief combination using the methods from this paper. We present two sets of experiments. The first uses a real web of trust, mined from Epinions. We wanted to see how *weighted average* compared with *maximum* for belief combination. We also wanted to see what quality of user population is necessary for the system to work well, and what happens if there are mixes of both low and high quality users. Finally, these methods would have little practical use if we required that users are perfect at estimating trusts of their neighbors, so we examine the effect that varying the quality of trust estimation has on the overall accuracy of the system.

For the second experiment, we implemented a real-world application available over the web (BibServ, http://www.bibserv.org). BibServ provides us with both anecdotal and experimental results. We first begin with the synthetic experiments

6.1 Epinions Web of Trust

We used the Epinions (www.epinions.com) web of trust for our network. Epinions in a product review site in which the reviews are provided by users. In order to maintain quality, Epinions encourages users to specify which other users they trust, and uses the resulting web of trust to custom-tailor the order of product reviews seen by each person. With over 75k users and 500k edges[3], Epinions' web of trust is an ideal source for web of trust experiments.

6.1.1 Experimental setup

We did not use the actual text reviews from Epinions since the Semantic Web deals with information, represented in our model by statements with associated beliefs. We instead imagined that the statements being combined were all of the form "Product attribute *a* has value *x*", where *x* is one of two values. We generated a ground truth by randomly assigning actual values to each attribute, which results in a set of M = 2·(number of attributes) statements, half of which are true. We expected that in a real-world system, the quality of information and trusts would vary from user to user, which we model by assigning a *quality* to each user. Each user asserts some number of statements, and the probability that an asserted statement is true is equal to the user's quality.

The Epinions web of trust is Boolean, but our methods work on real-valued trusts. We assume that a user with low quality is also bad at estimating trust, so we set the trust between two users to be the quality of the trusted user plus some uniformly distributed noise, which ranges +/- (1 – the quality of the trusting user).

Example: User A has a quality of 0.3, user B has a quality of 0.8. On average, 30% of user A's assertions are be true (70% false), likewise for B. If, in the Epinions web of trust, there was a link from A to B, then the trust between them would be some random value chosen uniformly from [0.1,1.0] (0.8 +/- 0.7). If there was a link from B to A then the trust would be randomly selected from [0.1, 0.5].

Each user's task is to select a set of attribute-value pairs which they believe best represents the ground truth. Our metric of performance is *precision*, which is the fraction of the pairs which are actually in the ground truth. As such, after calculating her combined beliefs, each user performs a final post-processing step to make them consistent. If the user holds a positive belief for both values of a single attribute, only the one with highest belief value will be kept (the other is set to 0).

The number of statements made by a user is equal to the number of Epinions reviews that user wrote. Users with high connectivity tend to have written more reviews, and there are few users that have written many reviews and many users that wrote few or no reviews. These characteristics are the same as we would expect to find in the distribution of statements in the network of trust for the Semantic Web.

Parameters

Unless otherwise specified, we used 5000 attributes, and the quality of a user was chosen from a Gaussian distribution with $\mu = 0.5$ and $\sigma = 0.25$, and λ is 0.5. A user's personal belief in any statement he asserted is 1.0, and is 0.0 for any other statement.

6.1.2 Comparing combining functions

In Table 1, we give results for a variety of belief combination functions. Let S_i be the set of statements s for which $\mathbf{B}_i > 0$. Let G be the set of actually true statements asserted by at least one user in the system[4]. Then:

$$\text{Precision}_i = \frac{|S_i \cap G|}{|S_i|}$$

Notice that the typical notion of recall is hard to define in this domain because we are dealing not just with the retrieval of information, but also whether or not that information was correct. To keep the metric of precision fair, our results report the precision at the highest possible recall. In other words, beliefs did not have to cross a threshold in order to be reported. We also found that for two reasonable definitions of recall, the results were meaningless anyway, as it depended on the structure of the network rather than the belief combination metric being used.

[3] The trust relationships can be obtained by crawling the site, as described in our previous work [38]. Though the full graph contains 75k users, we restrict our experiments to the first 5k users (by crawl-order), which forms a network of 180k edges.

[4] Note $S_i \cap G$ is the set of statements that the user correctly believed were true.

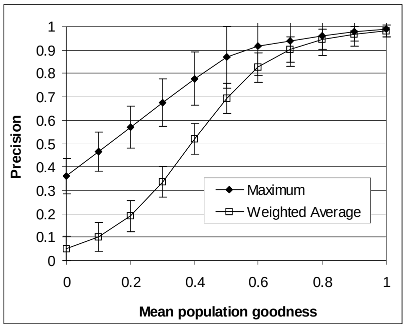

Figure 3: Average precision for *maximum* and *weighted average*. Error bars show one standard deviation on either side of the mean. Maximum has a statistically significantly (p<0.01) higher precision than normalized sum on this data set.

The combination functions *maximum* and *weighted average* are the same as introduced earlier. The first two "combination functions" in Table 1 are as follows: Combination function *None* is no trust combination (i.e. **T**=**t** and **B**=**tb**) – meaning that a user incorporates only the personal beliefs of his neighbors. With "combination function" *random*, $\mathbf{T}_{ij}$ is chosen uniformly randomly from [0,1], meaning that a user incorporates all of the beliefs of all users in the network indiscriminately. As expected, the precision for such a user is low. Since the average quality is 0.5, half of the facts in the system are true, so believing a random set of those facts leads to a precision of 0.5.

Weighted average and *maximum* outperform the baseline combination functions. Further, *maximum* significantly out performs *weighted average* in precision. Interestingly, the precision varied only slightly for users with high quality compared to users with low quality. We suspect this is because even a user with low quality may have good combined beliefs if all of his neighbors have good combined beliefs.

6.1.3 Varying the population quality

From the experiences of CiteSeer and OpenMind, we may expect the average quality of users to be fairly high [personal communication], but it is still important to understand how the average precision is affected by varying the quality of users.

We first explored the effect of varying μ, the mean population quality (see Figure 3). *Maximum* significantly outperforms *normalized sum* ($p < 0.01$). For high population quality, the two perform similarly, but with low quality, maximum's precision is considerably better than normalized sum produces considerably better precision results.

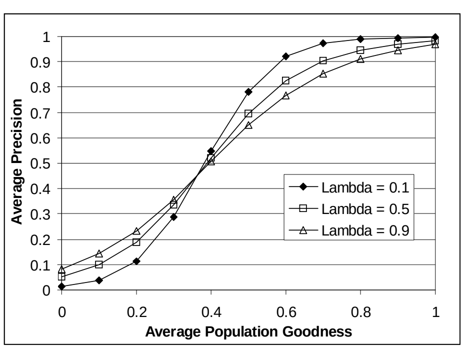

Figure 4: Effect of λ on the precision when combining with *weighted average*.

Table 1: Average precision and recall for various belief combination functions. The +/- value is the standard deviation.

Comb. Function	Precision
None	0.57 +/- 0.13
Random	0.5
Weighted Average	0.69 +/- 0.06
Maximum	0.87 +/- 0.13

Why does this happen? We believe it is because at low μ, there are few users with high quality. Those that do have high quality will be well trusted on average, but the trust in them is likely to be very noisy due to the overall low average quality. With *maximum* combination, each user needs just *one* path of high trust between themselves and the good user. With *weighted average* combination, the few high trust paths are overwhelmed by the significantly larger number low trust and simply noisy paths, resulting in too much uncertainty. Another way to look at it is that *maximum* filters out all noise by considering only the most trusted opinion, while *weighted average* incorporates all opinions. When the average quality of the users is high, the two should be similar, but when the average quality is low, *maximum*'s ability to filter information allows it to produce consistently better results.

We also wanted to explore the effect of varying λ for *weighted average*. In Figure 4, we see that λ has only a small effect on the results. Interestingly, high λ is best when the population is bad, while low λ is best when the population is good. Intuitively, the better the population, the more a user should consider their opinion, hence the lower λ should be. Because *maximum* seems to consistently outperform *weighted average*, and has the additional advantages of being cycle-indifferent and producing absolute beliefs, we restrict the remaining experiments to it.

6.1.4 Good and bad users

The distribution of quality so far has been Gaussian. To measure the robustness of the network to malicious (or just clueless) users, we varied the population by drawing from two Gaussian distributions, with means of 0.25 and 0.75 (both have the same standard deviation as earlier, 0.25). Users drawn from the first Gaussian are referred to as *bad*, while those from the second are *good*. In Figure 5 we see the effect of varying the fraction of *good* users in the network.

We found the network to be surprisingly robust to bad users. The average precision was very high (80-90%) when only 10-20% of the users were good. Consider also the situation where half of the users are good, which means the average quality of the system is 0.5. We can compare the precision of this network with the single-Gaussian results presented in Figure 3, which also have a mean quality of 0.5. Interestingly, the population consisting of good and bad users has a higher precision. This shows that it is more useful to have a few very good users than to have many mediocre ones.

These results are actually conservative. It is likely that the Epinions users who are more trusted, and who have written more reviews, are more likely to be good users.

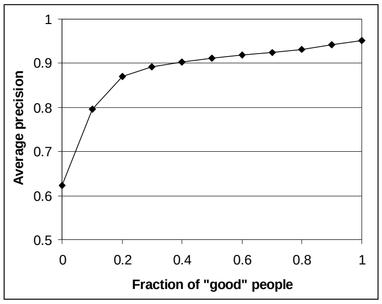

Figure 5: Precision for various fractions of good people in the network, using *maximum* belief combination.

6.1.5 Varying trust estimation accuracy

How well does the network handle inaccurate trust estimation? In this section we investigate how accurate the trusts must be in order to maintain good quality beliefs. For this experiment, we let the amount that user *i* trusts user *j* depend on a noise parameter δ:

$$\mathbf{t}_{ij} = \text{uniformly chosen from } [\max(\gamma_j-\delta, 0), \min(\gamma_j+\delta, 1)]$$

where γ_j is the quality of user *j*. Note that when $\delta=0$, $\mathbf{t}_{ij}=\gamma_j$, and when $\delta=1$, $\mathbf{t}_{ij}$ is chosen uniformly from [0,1].

Figure 6 shows the average precision for various values of δ. Clearly, if all of the trust values are random, and there is no correlation between the number of trusts to a person and the quality of their information, then the users will have completely random beliefs. Thus, the lowest precision is 0.5, when the noise is 1. The results demonstrate that the network is robust to noise, maliciousness, and low quality users. They also show that, for this particular domain, *maximum* outperforms *weighted average*. We now describe our BibServ system, and the real-world experiments it was used to conduct.

6.2 BibServ System

We have implemented these belief and trust combination functions in the BibServ system, the beta version of which is publicly accessible at www.bibserv.org. BibServ is a bibliography service which allows users to upload and maintain bibliographies, create new bibliographic entries, and edit, rate, and/or find any entry in the system, whether created by them or not.

6.2.1 Why bibliographies?

Before giving specific details on the implementation and application of belief combination to BibServ, we first motivate the choice of bibliographies as a domain. We felt that bibliographies have many characteristics that make them an excellent starting point for research into mass collaboration for the Semantic Web. Currently, because they are not connected, many people create bibliography entries for the same paper The result is a massive duplication of effort, and the creation of entries that may be inconsistent (due to errors) and may be of widely varying quality. Further, different people are likely to be interested in different subsets of all papers, so there are also issues of relevance to be handled. The bibliography domain is simple, yet gives rise to all of the issues of information quality, relevance, inconsistency, and redundancy that we desire to research. There are pragmatic reasons as well. Even without handling any of the complexities outlined above, a collection of bibliography entries is immediately useful, especially to researchers. The BibServ beta site currently has 70 users and 18000 bibliography entries[5], drawn mainly from the UW computer science department and IBM Almaden. We have received highly positive feedback from multiple users.

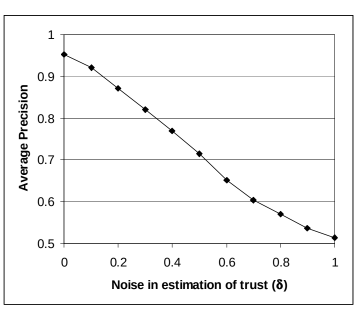

Figure 6: Effect of varying the quality of trust estimation.

The relation of BibServ to belief combination is as follows: BibServ users are presented with a list of other BibServ users whom they may rate for "trust and relevance". This forms the trust matrix **t**. Users are allowed to upload and create new bibliography entries. We treat each entry as a statement. Users may set their beliefs explicitly by rating the quality of the entry, and we implicitly assume a belief of 1.0 for any entry in their personal bibliography (unless otherwise explicitly rated). This forms the vector **b** for each entry. We currently use the weighted average combining function for merging beliefs and trusts.

6.2.2 Implementation

Because BibServ is implemented with a central server architecture, we chose to store the merged trusts **T** and compute the merged beliefs as needed. This requires O(NM) storage space. Since the number of users is much less than the number of bibliography entries, this is much less than the $O(M^2)$ space that would be required if we instead stored the merged beliefs.

By our definition, a user's merged belief in a bibliography entry represents the quality and relevance of that entry to them. Hence, when a search query is issued, we calculate the belief in each matching entry and present the search results ordered by belief. Incorporating both traditional measures of query relevance (for instance, TFIDF) and belief would likely lead to a better ordering of entries. One probabilistic-based technique for this is that of query-dependent PageRank [39], which we believe would fit nicely into the existing framework for belief merging. The computation of both trust and belief are implemented in SQL and, in the case of beliefs, is incorporated directly into the search query itself. The overhead of computing beliefs is typically less than 10% of the time required to perform the query itself. This is partially due to the fact that the number of beliefs in any particular entry is small, so computing the merged beliefs takes time approximately linear in the number of entries.

6.2.3 Experiments and results

We are still collecting data from BibServ, which has been "live" for only three weeks. The results presented in this section may not be reliable because they are based on a small quantity of data, but we believe that the trends indicated here are indicative of the results we will find as BibServ grows. We are continuing to perform experiments, and plan to report more extensive results in the camera ready version of this paper. Experiments were performed using weighted average or maximum as belief combination functions. Unless otherwise specified, $\lambda=0.5$ for weighted average. The web of trust specified by the BibServ users seems to follow a power-

[5] Aside from the half a million entries it was seeded with

law [8] distribution. Both the in-degree and out-degree of users have a R^2 value of over 0.8 with the power-law trend line.

Belief as quality and relevance

Our claim is that beliefs indicate the quality and relevance of bibliography entries to a particular user. To validate this claim, we ran the following experiment: We asked the pool of BibServ users to think of a particular paper they would like to look for, such as one they are currently reading, and use BibServ to search for a bibliography entry for that paper. We returned the search results in random order, and asked the user to rate each entry in terms of quality (0-5) and relevance (either "yes, this is the paper I was looking for" or "no, this is not". We also allowed 'It is hard to tell' to be used if necessary). Since we wished to compare beliefs with ratings for each search, we required that the search return at least 5 entries, and that the user rate every entry. The data set we have gathered so far consists of 405 ratings of quality and relevance by 13 different users, split across 26 searches.

The results of the experiment were positive, even if not inconclusive. We measured the correlation between belief and either quality or relevance. Correlation was measured for a data set consisting of all ratings and beliefs for all searches. Ratings and beliefs were first converted to z-scores on a per-search basis in order to remove user variation in rating interpretation. The best correlation was weighted average, which produced beliefs which had a correlation of 0.29 with the quality ratings (λ=0.03). The other correlations where 0.10 (weighted average vs. relevance), 0.16 (maximum vs. quality), and -0.01 (maximum vs. relevance). We expect that as BibServ grows, and there is a much wider diversity of users (currently, nearly all users are from computer science, and many specialize in data mining), that correlation, especially with relevance, will improve.

In order to measure the usefulness of beliefs for ordering search results, we measured the ratio of the average rating of the top k results (ordered by belief) vs. the average rating of all results, on a per search basis. The average ratio (across different searches) for relevance ranges from 1.2 to 1.6 for a variety of k (1-5) and for either belief combination function. The average ratio ranges from 0.96 to 1.05 for quality. The ratio rapidly tends towards 1.0 as k increases, indicating that while belief is a good indicator for relevance, the data contains a lot of noise (making it possible only to identify the very best few entries, not order them all).

Perhaps the most interesting result of the experiments was with regard to λ. We found that the best results when measuring beliefs vs. quality ratings were when λ was very small, though still non-zero. On the other hand, the best results when measuring beliefs vs. relevance were when λ was very large, though not equal to one. This indicates two things: 1) The majority of users share a similar metric for evaluating the quality of a bibliography entry, since personalizing the estimation of quality only degraded the results, and 2) Users have a widely varying metric for evaluating the relevance of a bibliography entry, which is intuitive since users may all have different specialties. The fact that the best λ was not 0 or 1 indicates that in both cases, both information gleaned from others, as well as having personalized beliefs, can be useful.

7. RELATED WORK

The idea of a *web of trust* is not new. As mentioned, it is used by Epinions for ordering product reviews. Cryptography also makes use of a web of trust. In public-key cryptography, if John wishes to send a secret message to Jane then he must first get Jane's public key. The difficulty is, how can he verify that the key he receives actually belongs to Jane? The standard approach depends on *certificate authorities* to verify user identification. Interestingly, a grassroots decentralized approach (PGP) has also arisen, which essentially allows John to verify Jane's public key through a network of trusted users. Discussion and extensions of this can be found in [11]. In Abdul-Rahman's distributed system for trust management, John's trust in Jane, and John's trust in Jane's ability to determine who is trustworthy, are kept explicitly separate, though trusts are discrete and only qualitatively valued [1]. We think such a separation would be interesting to consider in our framework as well.

The analogy to belief combination on the web is ranking the quality of web pages. Information retrieval methods based on the content of the page (such as TFIDF [26]) performed reasonably, but considered each page to be independent. When methods were developed which also considered the connectivity between pages [33][13][29], web page quality estimation improved dramatically. We see our belief combination on the Semantic Web as analogous to such methods on the WWW

There have been a number of previous projects on mass collaboration. Quiq is one deployed application of mass collaboration which has been successful. OpenMind (www.openmind.org) aims to collect large amounts of information, most typically training data such as handwriting and speech, through mass collaboration. The Open Directory Project (www.dmoz.org) aims to build a taxonomy of web pages through volunteer effort. However, none of these projects explicitly incorporate a concept of trust, that we are aware of. These projects, and indeed any mass collaboration domain, may be able to benefit from the web of trust calculations presented in this paper.

In previous work [18], we mined trust relationships from online sites, and applied social network algorithms to them in order to identify users with high network influence. Applying the same methods to the Semantic Web web of trust may prove fruitful in identifying useful contributors, highly respected entities, etc. Also in a similar vein is the ReferralWeb project, which mines multiple sources to discover networks of trust among users [27]. Also interesting is collaborative filtering [37], in which a user's belief is estimated by considering the beliefs of users he is similar to. This can be seen as belief combination without explicit trusts, or even as forming the web of trust implicitly, based solely on similarity of interests.

Work on belief combination has had a long history. Fuzzy logic [41] and certainty factors [5] are just two of the fields of study devoted to the topic. A summary of various belief combination functions can be found in [20] and [22]. For more a probabilistic treatment, see [35]. A different form of belief combination is that of Pennock et. al [36] who looked at how web-based artificial markets may combine the beliefs of their users.

There are a number of probabilistic approaches to belief combination that would be interesting to explore. Since our model is network-based, we believe that graphical models such as Bayesian networks [35], Markov random fields [14], and/or dependency networks [25] could be used for belief combination. Presently, such models are too computationally expensive to scale to networks of the scale of the Semantic Web.

As mentioned in Section 2.1, much work has been done on inference methods for statements augmented with beliefs or probabilities. Knowledge-based model construction (KBMC) [32] and related methods [28][30] augment first-order rules with probability and describe how inference is then performed. Bayesian networks are used to determine beliefs when the information is in the form of conditional probabilities for propositional variables. Similarly, probabilistic relational models [21] (PRMs) are used when the probabilities concern relational statements. We consider such work to be complementary to our belief combination, in that both are needed for solving separate aspects of the problem of beliefs on the Semantic Web.

8. Future work

In this work, we assumed that statements are independent. We would like to investigate how dependencies between statements, with varying beliefs on each, may be handled. For example, if we consider a taxonomy to be a set of class-subclass relationships, and consider each relationship to be an independent statement, then merging such taxonomy beliefs is likely to lead to a nonsensical taxonomy. We would like to be able to merge structural elements like taxonomies, and expect that approaches such as [16] and [17] may provide useful insights into possible solutions.

The path algebra and probabilistic interpretations were shown to be nearly identical, and the probabilistic interpretation is a generalization of PageRank. Considering PageRank works so well on web pages, it would be interesting to apply the ideas developed here back to the WWW for the purposes of ranking pages. For instance, might we find it useful to replace the sum with a maximum in PageRank?

We would like to consider networks in which not all users employ the same belief combination function. Can we guarantee any kind of global interpretation in such a case? In the path algebra formulation, it appeared that in order to find a good global interpretation, we would need the aggregation and concatenation functions to have overly-restrictive properties. We would like to investigate modifying the global interpretation in order to relax the requirements.

There are many tradeoffs between computation, communication, and storage requirements for the different architectures (peer to peer, central server, hierarchical, etc.), algorithms (semi-naïve, Warshall, etc.), and strategies (merge beliefs on demand, store all beliefs, etc.). We would like to formalize these tradeoffs for better understanding of the efficiency of the various architectures.

In this paper, we considered only single valued beliefs and trusts. In general, a belief could actually be multi-valued, representing a magnitude in multiple dimensions, such as ‘truth’, and ‘importance’, and ‘novelty’. We would also like to consider multi-valued trusts, which may represent similar dimensions as beliefs (but applied to users). It may be possible to combine beliefs and trusts into one concept, ‘opinion’, which may be similarly applied to both statements and users.

Similarly, we would also like to allow users to specify *topic-specific* trusts. We expect that with topic-specific trusts, the normalized sum combination function would be similar to calculating query-dependent PageRanks [39]. We would also like to apply query-dependent PageRank to the problem of merging a statement’s belief and a statement relevance for the purposes of presenting a user with an ordered list of statements that are both relevant and highly believed.

9. CONCLUSIONS

If it is to succeed, the Semantic Web must address the issues of information quality, relevance, inconsistency and redundancy. This is done on today's Web using algorithms like PageRank, which take advantage of the link structure of the Web. We propose to generalize this to the Semantic Web by having each user specify which others s/he trusts, and leveraging this "web of trust" to estimate a user's belief in statements supplied by any other user. This paper formalizes some of the requirements for such a calculus, and describes a number of possible models for carrying it out. The effectiveness of the approach, and the tradeoffs involved, are illustrated by experiments using data from the Epinions knowledge-sharing site, and from the BibServ site we have set up for collecting and serving bibliographic references.

10. ACKNOWLEDGEMENTS

We thank Ramanathan Guha for discussions on the Semantic Web and networks of trust. We also thank James Lin for his help with BibServ’s site design.

11. REFERENCES

[1] A. Abdul-Rahman and S. Hailes (1997). A distributed trust model. Proceedings of New Security Paradigms Workshop.

[2] R. Agrawal and H.V. Jagadish (1988). Multiprocessor transitive closure algorithms. In *Proc. Int’l Symp. on Databases in Parallel and Dist. Systems*. Austin, Texas, Dec. 5-7, 1988.

[3] R. Agrawal, S. Dar, and H. V. Jagadish (1990). Direct transitive closure algorithms: Design and performance evaluation. *TODS*, 15(3):427-458.

[4] A. V. Aho and J.D. Ullman (1975). *The design and analysis of computer algorithms*. Addison-Wesley, Reading, MA.

[5] G. P. Amata Cruz and Gleb Beliakov (1996). On the interpretation of certainty factors in expert systems. *Artificial Intelligence in Medicine*, 8(1):1-14.

[6] A. Ankolekar et al. (2002), "DAML-S: Web Service Description for the Semantic Web," Proc. *1st Int'l Semantic Web Conf*. (ISWC 02), 2002.

[7] F. Bancilhon (1985). Naïve evaluation of recursively defined relations. *Tech Report DB-004-85*, MCC, Austin, Texas.

[8] A. L. Barabási, R. Albert, and H. Jong (2000). Scale-free characteristics of random networks: The topology of the World Wide Web. *Physica A*, 281:69-77, 2000.

[9] R. Bellman and M. Giertz (1973). On the analytic formalism of the theory of fuzzy sets. *Information Sciences*, 5:149-156.

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

[11] M. Blaze, J. Feigenbaum, and J. Lacy (1996). Decentralized trust management. In *Proc. of the IEEE Symposium on Research in Security and Privacy*, Oakland, CA, May 1996.

[12] B. Carre (1978). *Graphs and Networks*. Clarendon Press, Oxford, England.

[13] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan (1998). Automatic resource compilation by analyzing hyperlink structure and associated text. In *WWW*, pages 65-74, Brisbane, Australia, 1998. Elsevier.

[14] R. Chellappa and A. K. Jain, editors (1993). *Markov random fields: theory and application*. Academic Press, Boston, MA.

[15] D. M. Chickering and D. Heckerman (1997). Efficient approximations for the marginal likelihood of Bayesian net-

works with hidden variables. *Machine Learning*, 29:181-212, 1997.

[16] A. Doan, J. Madhavan, P. Domingos, and A. Y. Halevy. Learning to map between ontologies on the semantic web. In *WWW*, pages 662-673, 2002.

[17] A. Doan, P. Domingos, and A. Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. In *Proc. of the 2001 ACM SIGMOD Inter'l Conf. on Man. of Data*, pages 509-520, Santa Barbara, CA, 2001. ACM Press.

[18] P. Domingos and M. Richardson (2001). Mining the network value of customers. In *Proc. KDD 2001*.

[19] B.J. Fogg, L. Marable, J. Stanford, and E. Tauber (2002). *How do people evaluate a web site's credibility? Results from a large study*. Stanford Persuasive Technology Lab, Stanford, CA 94305.

[20] S. French (1985). Group consensus probability distributions: A critical survey. *Bayesian Statistics* 2, pp.183-202.

[21] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer (1999). Learning probabilistic relational models. In *Proc. IJCAI 1999*.

[22] C. Genest and J. V. Zidek (1986). Combining probability distributions: A critique and an annotated bibliography. *Statistical Science* 1(1) pp.114-148.

[23] A. Y. Halevy, Z. G. Ives, D. Suciu, and I. Tatarinov (2003). Schema mediation in peer data management systems. *ICDE*.

[24] T. H. Haveliwala (2002). Topic-sensitive pagerank. In *Proc. WWW 2002*.

[25] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie (2000). Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research* vol. 1, pp.49-75.

[26] T. Joachims (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization.*ICML-97*.

[27] H. Kautz, B. Selman, and M. Shah. ReferralWeb: Combining social networks and collaborative filtering (1997). *Communications of the ACM*, 40(3):63-66.

[28] K. Kersting, L. De Raedt, and S. Kramer (2000). Interpreting Bayesian logic programs. In *Proc. AAAI-2000 Workshop on Learning Stat. Models from Rel. Data*.

[29] J. M. Kleinberg. Authoritative sources in a hyperlinked environment (1998). In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*.

[30] D. Koller and A. Pfeffer (1997). Learning probabilities for noisy first-order rules. In *Proc. IJCAI 1997*

[31] R. Motwani and P. Raghavan (1995). *Randomized Algorithms*. Cambridge University Press, United Kingdom, 1995.

[32] L. Ngo and P. Haddawy (1997). Answering queries from context-sensitive probabilistic knowledge bases. *Theoretical Computer Science*, 171:147-77.

[33] L. Page, S. Brin, R. Motwani, and T. Winograd (1998). The PageRank citation ranking: Bringing order to the web. Technical report, Stanford University, Stanford, CA.

[34] Peter Patel-Schneider and Jerome Simeon (2002). Building the Semantic Web on XML. In *Int'l Semantic Web Conf.*

[35] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, CA, 1988

[36] D. M. Pennock, F. A. Nielsen, and C. L. Giles (2001). Extracting collective probabilistic forecasts from Web games. In *Proc. ACM SIGKDD 2001* pp.174-183, San Francisco, CA.

[37] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews (1994). In *Proc. CSCW 1994*

[38] M. Richardson and P. Domingos (2002). "Mining Knowledge-Sharing Sites for Viral Marketing," *SIGKDD 2002*

[39] M. Richardson and P. Domingos (2002). The intelligent surfer: Probabilistic combination of link and content information in PageRank. In *NIPS* 14. MIT Press.

[40] S. Warshall (1962). A theorem on Boolean matrices. In *Journal of the ACM* 9(1):11-12, 1962.

[41] H.-J. Zimmerman (1996). *Fuzzy set theory and its applications*. Kluwer Academic Publishers, Norwell, MA.

12. APPENDIX

In this Appendix, we give a proof of Theorem 1. From the theorem, we may assume that ◊ is commutative and associative, and ○ is associative and distributes over ◊, and **t**, **T**, **b**, and **B** are defined as in Section 3. Also from Section 3,

$(\mathbf{A}\bullet\mathbf{B})_{ij}=\Diamond(\forall k\colon \mathbf{A}_{ik}\circ\mathbf{B}_{kj})$

We first prove the following:

***Lemma:* • is associative**

Let $\mathbf{X}=(\mathbf{A}\bullet\mathbf{B})\bullet\mathbf{C}$

Then

$$
\begin{aligned}
\mathbf{X}_{ij} &= \Diamond(\forall k\colon \Diamond(\forall l\colon \mathbf{A}_{il}\circ\mathbf{B}_{lk}) \circ \mathbf{C}_{kj}) && \text{from definition of } \bullet \\
&= \Diamond(\forall k\colon \Diamond(\forall l\colon \mathbf{A}_{il}\circ\mathbf{B}_{lk}\circ\mathbf{C}_{kj})) && \text{since } \circ \text{ distributes over } \Diamond \text{ and } \circ \text{ is associative} \\
&= \Diamond(\forall l\colon \Diamond(\forall k\colon \mathbf{A}_{il}\circ\mathbf{B}_{lk}\circ\mathbf{C}_{kj})) && \text{since } \Diamond \text{ is associative} \\
&= \Diamond(\forall l\colon \mathbf{A}_{il}\circ \Diamond(\forall k\colon \mathbf{B}_{lk}\circ\mathbf{C}_{kj})) && \text{since } \circ \text{ distributes over } \Diamond \\
&= \Diamond(\forall l\colon \mathbf{A}_{il}\circ (\mathbf{B}\bullet\mathbf{C})_{lj}) && \text{by definition of } \bullet \\
\Rightarrow \quad \mathbf{X} &= \mathbf{A}\bullet(\mathbf{B}\bullet\mathbf{C}) && \text{by definition of } \bullet.
\end{aligned}
$$

Belief Merging:

We have

$$\mathbf{B}^{(0)} = \mathbf{b} \text{ and } \mathbf{B}^{(n)} = \mathbf{t}\bullet\mathbf{B}^{(n\text{-}1)}$$

So, $\mathbf{B}^{(1)} = \mathbf{t}\bullet\mathbf{b}$, $\mathbf{B}^{(1)} = \mathbf{t}\bullet\mathbf{b}$

or $\mathbf{B}^{(n)} = \mathbf{t}\bullet(\ \mathbf{t}\bullet(\ldots \bullet(\ \mathbf{t}\bullet\mathbf{b}))))\ldots$

From the lemma, • is associative, so we find

$$\mathbf{B}^{(n)}= \mathbf{t}^{n}\bullet \mathrm{b} \tag{8}$$

(where $\mathbf{t}^{n}$ means $\mathbf{t}\bullet\mathbf{t}\bullet\mathbf{t}\ldots$ n times, and $\mathbf{t}^{0}$ is the identity matrix)

Trust Merging:

We have

$$\mathbf{T}^{(0)} = \mathbf{t} \text{ and } \mathbf{B}^{(n)} = \mathbf{t}\bullet\mathbf{T}^{(n\text{-}1)}$$

$$\mathbf{T}^{(n)} = \mathbf{t}\bullet(\ \mathbf{t}\bullet(\ldots \bullet(\ \mathbf{t}\bullet\mathbf{t}))))\ldots$$

$$\mathbf{T}^{(n)}= \mathbf{t}\bullet \mathrm{t}^{n} \tag{9}$$

Theorem:

Combining Equations 8 and 9,

$$\mathbf{t}\bullet\mathbf{B}^{(n)} = \mathbf{T}^{(n)}\bullet\mathbf{b}$$

Since we run each recursion until convergence, this is sufficient to show that

$$\mathbf{t}\bullet\mathbf{B} = \mathbf{T}\bullet\mathbf{b}$$