

Trusting Claims from Trusted Sources: Trust Network Based Filtering of Aggregated Claims

Jennifer Golbeck¹, Bijan Parsia¹

University of Maryland, College Park, MINDSWAP, 8400 Baltimore Avenue
College Park, MD 20742 USA
golbeck@cs.umd.edu, bparsia@isr.umd.edu

Abstract. On the semantic web, assertions may be aggregated from many sources, those aggregations filtered, reasoned over, aggregated with other aggregators, displayed, scraped, extracted, recombined, and otherwise processed without significant human oversight. To preserve the connection between assertions and their source, various provenance schemes for semantic web data have been explored. However, the primary focus has been on authenticating the author of a particular statement, e.g., using digital signatures, but there is no provision for relating the authenticity of the source of the assertion and the trustworthiness of the assertion itself. This paper presents a method for using semantic web based trust networks to infer the reputation of sources for a statement and compose the reputation of several sources. By calculating a trust rating for each statement based on the ratings of its sources, the set of statements can be filtered based on the rating.

1 Introduction

Information – in particular, "content" – on the World Wide Web is presented with an expectation that the information consumer is a human being. People are expected to make use of a variety of cues to ascertain, for example, the proponent of a claim, the author of an article, or the photographer who took a photo. Most of these cues are traditional: bylines, attributions, quotations, citations, authorial claims, copyright notices, and the like. Some cues derive from features of the Web architecture, such as the use of the Domain Name System (DNS) in, for example, HTTP URIs, or HTTP redirects. Digital signatures can be used to verify the particular origin of a document, and that the document was unchanged in transit, but there is no provision for relating the authenticity of the source of the document and the trustworthiness of the content of that document. Human judgment is required to determine the nature of the document and its content (e.g., real purchase order, example order for debugging, or a parody for amusement). One way that the need for continual human intervention can be eliminated is for people and organizations to set up agreements that certain documents exchanged in certain contexts will be reliable in the appropriate ways. Given that the parties of such agreements all trust each other, accepting information reduces to verifying that it came from a trusted source.

The Semantic Web is conceived as the "next generation" of the World Wide Web, wherein much of the content of the Web will not be solely, or even primarily, intended for human consumption. Instead, content sensitive programs will collect, process, exchange, generate, and make decisions based on Web accessible information. As Web agents make more significant decisions, it become more imperative that they are more sophisticated in how they accept information from the Web. While many Semantic Web programs will have significant domain knowledge and thus, presumably, some built-in methods for evaluating the plausibility of new information, perhaps the majority of them will be less specialized. Thus, there is a need for more general, not content specific, techniques.

Many websites are *open*, in that nigh anyone can submit information to be published on the site. This can range from very restricted submissions, such as comments on articles, to the entire content of the site, as with Wikis. This is relatively unproblematic when the information submitted is always presented as a cohesive chunk, say, a Wikipage, or a specific comment, or a specific blog entry. This permits the human reader to evaluate both the content of the chunk and the context of submission (i.e., the provenance).

In contrast, in an open *semantic* website, this is not sufficient. For example, <http://owl.mindswap.org/> is a open, RDF and OWL driven website. It accepts relatively arbitrary submission of bits of RDF and OWL. It incorporates the assertions in a submission in a variety of contexts, presenting the assertions in contexts divorces from their submission and using those assertions to draw inferences, which are themselves presented on different pages. The page generation software has to decide how and where to present or otherwise use each assertion in a submission.

In this paper, we present a method for integrating semantic web based trust networks with provenance information to rate and filter a set of assertions. We will describe an ontology for creating trust networks and present an accurate algorithm for inferring trust relationships. Those inferred values are then used to compose ratings of the reputation or trustworthiness of assertions. We describe how those ratings on assertions can then be used to filter the set of statements used in an application, thereby creating a knowledge base with a known level of validity.

2 Background and Related Research

2.1 Trust on the Semantic Web

The issue of trust and reputation on the web has been around since the web itself began. How users could have trust in websites has taken many forms. In an unpublished masters thesis [2], Dudek studied user trust in e-commerce systems, and found that it closely related to the aesthetics of the site. More formal methods for rating the reputation of a site or of a user are also common. The Ebay rating system tries to use customers positive and negative feedback ratings as a measure of a seller's reputation. Epinions, a consumer reviews website, also allows customers to rate the transactions with sellers, and maintains a more explicit trust rating system. The Epinions dataset is commonly used in the study of trust on the web. The PageRank

algorithm[14], used by the Google search engine, also is a trust metric of sorts. It uses the number of links coming into a particular page as votes for that site. This rating, combined with other text processing, is used to score results. The PageRank algorithm is so effective at rating the relevance of pages, that its results are commonly used as a control for testing the effectiveness of trust metrics.

Explicitly creating networks and analyzing them to form opinions about the reputation of content and users has been an issue gaining more attention over the last few years, particularly with the advent of the semantic web. Yolanda Gil and Varun Ratnakar addressed the issue of trusting content and information sources [4]. They describe an approach to derive assessments about information sources based on individual feedback about the sources. As users add annotations, they can include measures of Credibility and Reliability about a statement, which are later averaged and presented to the viewer. Using the TRELIS system, users can view information, annotations (including averages of credibility, reliability, and other ratings), and then make an analysis.

In the peer to peer context, the EigenTrust system [8] (based on PageRank) effectively computes global trust values for peers, based on their previous behavior. Individuals with poor performance will receive correspondingly low trust ratings. Their system was shown to be highly resistant to attack.

Raph Levin's Advogato project [10] also calculates a global reputation for individuals in the network, but from the perspective of designated *seeds* (authoritative nodes). His metric composes assertions from members to determine membership within a group. The Advogato website at <http://advogato.org>, for example, certifies users at three levels – apprentice, journeyer, and master. Access to post and edit website information is controlled by these certifications. Like EigenTrust, the Advogato metric is quite attack resistant. By identifying individual nodes as "bad" and finding any nodes that certify the "bad" nodes, the metric cuts out an unreliable portion of the network. Calculations are based primarily on the good nodes, so the network as a whole remains secure.

Less centralized metrics are presented in work by Golbeck [5] and Richardson, et al. [17]. These metrics both use a local system for inferring reputations or trust relationships from a network. Richardson's work, like EigenTrust, presents a probabilistic interpretation of global belief combinations. The effectiveness of the system was shown in context with Epinions and BibServ.

2.2 Provenance

The Web is an largely unregulated, shared information space. Publishing information on the Web is typically a matter of being delegated control over some set of URIs and ensuring that there is appropriate server behavior in response to requests involving those URIs. HTTP URIs, for example, typically invoke the Domain Name System (DNS) in order to determine the "publisher" or "server" provenance of a Web page. By looking up the hostname, a client can determine which server is the authoritative origin of *au courant* Web content associated with the range of URIs keyed off that hostname. Caches, for example, can verify that their cached copy of an HTML document is current by resolving the host name and using the HTTP protocol to

retrieve either more information about the document, or a current version of the document, or whatever successor content the current domain name owner is now providing. As it stands, the Web has no inherent mechanism for ascertaining publishing provenance over time. DNS lookup is not temporally indexed, and thus DNS servers do not provide historical information of registrants of domain names. Even if the registrant does not change, URIs do not encode any temporal information, so information about the evolution over time of the content accessible via a particular URI must be maintained out of band or as part of the content itself. There are several websites that expose databases of temporally and URI indexed Web content, providing a historical record of the changes to the content of a fragment of the Web. This does not solve the problem, though. The Internet Archive, for example, does not snapshot DNS registrants, so key publishing provenance is lost. However, these defects should not minimize the key role that DNS like provenance information is absolutely essential to the Web and, in fact, functions extraordinarily well.

Thus far, we have discussed various mechanisms for determining that the content we have at hand is as a determined supplier is committed to endorsing as the content actually provided. A digitally signed document is guaranteed to be unmodified from the time of signing. This means that we can determine that the content is unchanged between us and the supplier, but we cannot determine the further relationship between the content and the provider, e.g., whether the provider the (sole) author of a document. The Dublin Core Metadata Initiative's core metadata elements provide a both a conceptual framework and a set of concrete mechanisms for representing structured provenance information. For example, Dublin Core distinguishes between creator and publisher as well as providing techniques for representing the creator and publisher of a Web published document in HTML, XML, and RDF.

The Inference Web project [<http://www.ksl.stanford.edu/software/IW/>] at Stanford University's Knowledge Systems Laboratory aims to provide an infrastructure for explaining Semantic Web information in ways to increase users confidence in the information. Provenance is critical to explaining why the system supplied a certain answer. Inference Web distinguishes two sorts of provenance, *knowledge provenance* and *knowledge process information* [1]. *knowledge provenance* includes descriptions of the source or origin of the information and of the chain of custody. This encompasses both sorts of provenance information discussed above. *Knowledge process information* is "a description of the reasoning process used to generate the answer." While Inference Web is currently focus on answers generated from deductive reasoning processes, there's no reason to exclude descriptions of general data transformation and analysis processes as is, for example, supported by myGRID for bioinformatics computational experiment workflows[18].

3 Trust

Trust and reputation are both socially loaded words. To use the terms as a computational concepts requires a strong definition. The first cut to make is to differentiate trust in this work from security. There has been quite a bit of research

that uses trust in this sense, but this research focuses on trust as social concept, and the methods described here are not intended to be used as security measures.

There has been some work toward formalizing trust in the social sense as a computational concept. The work by Deutsch in 1962 [2] contains a widely used definition. He states that trusting behavior occurs when a person encounters a situation where he or she perceives an ambiguous path, the result of which can be good or bad, and the occurrence of the good or bad result is contingent on the action of another person. Additionally, the negative impact of the bad result is greater than the positive impact of the good result, so the person is motivated to make the correct choice. If this person chooses to go down the path, the person has made a trusting choice – the belief that the outcome will be good lies in the fact that the other person on whom the outcome depends is trusted to do the right thing. Aspects of this definition have been countered, but the underlying concept is present in much of the trust research; namely that trust is a measure of confidence in something or someone in the face of uncertainty. In a sense, how much a person trusts another is a measure of how certain (or uncertain) he or she is that the trusted person will act in a way that brings a good outcome in a given situation.

In this work, trust is treated as a measure of certainty in a person or resource. Specifically, given an ambiguous path as described above, having trust in a person is defined as a measure of the confidence that the person will take the action that leads to the positive result. Reputation is synonymous with the measure of that trust.

3.1 Building a Trust Network

In this work, the trust networks used for inferring the reputation of people making assertions is Semantic Web-based. The foundation of a trust network is social networks. The most popular mechanism for describing social networks on the Semantic Web is the Friend-Of-A-Friend vocabulary. The Friend-Of-A-Friend Project (FOAF) is an RDF/OWL ontology that provides a vocabulary for describing people and their relationships [15]. The analysis in this paper used the trust network developed as part of the Trust Project at <http://trust.mindswap.org>. This network uses an ontology [19] that extends FOAF and allows people to rate the reputation of other people. The ontology uses a scale from 1 to 10, where 1 represents low trust and 10 represents very high trust, and the network currently contains nearly 1,400 people, and almost 1,700 trust relationships. This network is large enough to provide a foundation for our analysis.

The ontology also provides the capability to create trust ratings with respect to a specific topic. For example, Person A may rate Person B highly with respect to computer science knowledge, but give a low rating to Person B with respect to movie recommendations. The subgraph of edges with the selected subject is extracted from the whole network, and utilized in the same way. In the context of provenance tracking, it is possible to extract the trust ratings related to the statements of interest, and use only those ratings in the analysis. This work uses general trust ratings, not context specific ones, because there are more general ratings in the network, and the algorithms presented here are the same for general and context specific ratings.

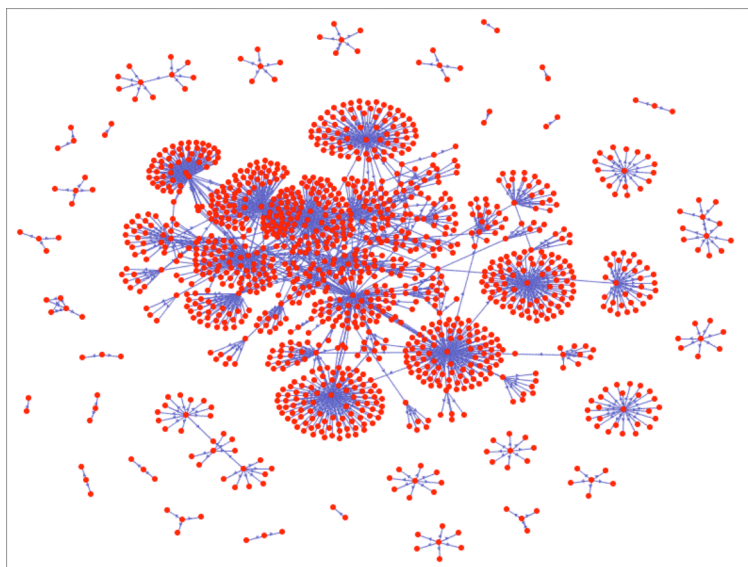


Fig 1. The trust network used in our analysis, developed at <http://trust.mindswap.org>

3.2 Perspective in Trust Metrics

In showing the potential utility of trust networks on the semantic web, a good mechanism is required for inferring trust ratings. A trust metric is used with the trust network and makes recommendations about how much one person should trust another. The trust literature contains many different metrics, each designed and implemented for specific purposes. These metrics can all be categorized according to the perspective they use for making calculations. *Global* metrics calculate a single value for each entity in the network. *Local* metrics calculate a trust rating for an individual in the network based on the ratings and preferences of the node that is the source of the query. In the global system, an entity will always have the same inferred rating. In the local system, an entity could be rated differently depending on which node the inference is made for.

The choice of which method is best depends on the context. Global metrics can be highly effective – Google’s PageRank algorithm uses a global metric for rating the relevance of web pages, and is highly effective. In peer-to-peer systems trust is used to rate individual sources, based on the source’s propensity to provide corrupted, incomplete, or invalid files. Since, in peer-to-peer systems, all users have access to essentially the same set of files and network features with a given source, a global metric makes sense – there is little evidence to suggest that one group of people would have a different experience than the general population when retrieving files from a source.

In a context where the experience of users vary, a local metric may be more appropriate. Local trust metrics use the neighbors and ratings of a particular node to make a trust inference. The network in figure 2 illustrates a simple example of why

the local metric is important in trust networks. Since neither A nor B have a trust rating for E, they must infer it. Each has a direct path through a single intermediate neighbor. Both A and B trust their respective neighbors at a level 10 – the highest trust rating. The neighbors, C and D, have very different opinions of E. Since A has high trust in C, and C has low trust in E, it should be more likely that A is recommended a low rating for E. On the other hand, B should receive a high trust recommendation for E since B has high trust for D, which in turn has high trust for E.

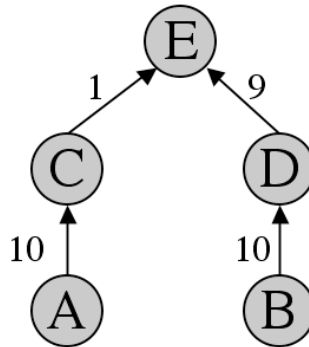


Fig. 2. Depending on local opinions, inferred values can be very different. In this graph, Node A and Node B should receive very different opinions of Node E because of the opinions of their highly trusted neighbors

3.3 Accurate Metrics for Inferring Trust

To analyze how accurately trust can be inferred over a network, we have created a simple metric. The inferred rating from the source to the sink is given by a weighted average of the neighbors' trust ratings of the sink. Notationally, the trust rating t from the source, i , to the sink, s , is written t_{is} .

If the source is directly connected to the sink, then no inference is necessary – the source already has a rating. If the two nodes are not directly connected, the source computes rating by calculating a weighted average of the trust ratings returned for the sink by each of its n neighbors. Each of the source's neighbors perform this same algorithm to find their ratings.

In the pseudocode below, lines 2 and 7 mark nodes as seen or unseen to prevent loops. Line 11 unmarks a neighbor so that if it is found in the path from another neighbor to the sink, its value can be used again. Line 12 caches the inferred rating from the source to the sink. Line 13 then returns the rating, either from the newly cached value, or the original rating from source to sink that already existed.

```

1  getRating(source, sink)
2  mark source as seen
3  if source has no rating for sink
4    denom = 0
5    num = 0
6    for each j in neighbors(source)
7      if j has not been seen
8        denom ++
9        j2sink =
          min(rating(source, j), getRating(j, sink))
10       num += rating(source, j) * j2sink
11       mark j unseen

12   rating(source, sink) = num/denom

13  return rating(source, sink)

```

Formula (1) shows the concise representation of how t_{is} is weighted. The rating from each neighbor is essentially multiplied by the rating of that neighbor, which is summed in the divisor. Consider the simple example where the source has two neighbors. Neighbor 1 is trusted at a level 1 (the lowest) and Neighbor 2 is trusted at a level 10 (the highest). Each neighbor is given a number of votes equivalent to its rating. Thus, Neighbor 1 receives one vote out of eleven (1 + 10), and Neighbor 2 gets ten votes out of eleven.

$$t_{is} = \frac{\sum_{j=0}^n \left\{ \begin{array}{l} (t_{js} * t_{ij}) \text{ if } t_{ij} \geq t_{js} \\ t_{ij}^2 \text{ if } t_{ij} < t_{js} \end{array} \right\}}{n} \quad (1)$$

The caveat is that the source will never trust the sink more than any intermediate node. For example, say the source has only one neighbor, and trusts that neighbor at a level 1. If the neighbor returns a value of 9 for the sink, should the source trust the

sink at a level 9? More precisely, should the source ever trust the sink more than the source trusts the intermediate party? Arguments can be made either way, but with the premise that trust is a measure of uncertainty, it is reasonable to enforce that uncertainty never decrease. Thus, in this weighting, the source takes the minimum of the neighbors rating and the neighbors recommendation about the sink. A single neighbor rated at a level 1 can make any recommendation, but the sink will never be trusted more than a level 1. If, on the other hand, a single neighbor with a rating of 10 recommends a rating of 9 for the sink, the source will accept that since it does not exceed the rating for the neighbor.

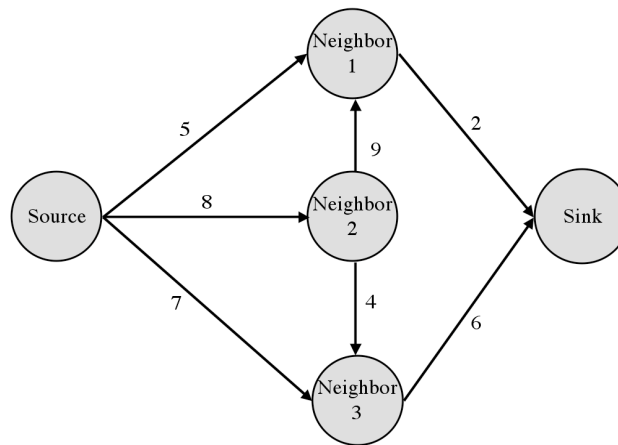


Fig. 3. This figure illustrates how the opinions of a single neighbor can be considered several times. In this case, neighbor 1 and neighbor 3 will each be considered directly by the source and also in the rating determined at neighbor 2

The figure above shows the case where a node's rating can be counted twice. The source will make calculations based only on its local view of the world. The ratings of Neighbors 1, 2, and 3 will be given their respective weight in the calculation. However, Neighbor 2 is not directly connected to the sink. Its rating is inferred from the ratings of Neighbors 1 and 3. Even though the source has already considered Neighbors 1 and 3 directly, this algorithm does not prevent them from being used again down the line. This issue is worth investigating further, and is discussed in the future work section.

3.4 Trust Metric Evaluation

Thus far, a simple metric over continuous values has been presented, and justified by the assertion that local metrics are better suited to trust inferences than global metrics. To support this claim, we implemented several metrics and tested them within the trust network developed as part of this project.

The primary objective in this experiment was to determine the accuracy of each metric. The experiment was performed by iterating through each individual i , in the network. The trust rating, t_{ij} , for each neighbor, j , was recorded. Then, the connection from i to j was removed. Using the rest of the network and selected metric, a trust rating, t_{ij}' , was inferred. The accuracy of each inference was measured as $|t_{ij} - t_{ij}'|$.

The control set of inferences were calculated by always setting t_{ij}' to the average trust rating in the network. The average difference between t_{ij} and t_{ij}' in the control was 1.74. When compared to the ten possible values of trust, this is a 17.4% difference, or an accuracy of 82.6%.

To see the benefits of trust inferences, several metrics were implemented and their $|t_{ij} - t_{ij}'|$ values were compared to the control using a standard, 2-tailed t-test. When the weighted average metric from above was implemented, it significantly outperformed the control ($p < .001$). The average difference between the actual and inferred rating was only 1.16 – an accuracy of 88.4%.

Table 1. Results of experiments to determine the accuracy of trust metrics

	Control: Average Rating	Weighted Average	Global: Authoritativ e Node	Global: Average ratings Assigned to the sink
$ t_{ij} - t_{ij}' $	1.74	1.16	1.459	1.487
Std. Dev.	0.95	1.21	1.45	1.49
Accuracy	0.826	0.884	0.8541	0.8513

Several global trust metrics were also implemented in the experiment. The first global metric was identical to the local metric above except that it always used the same node as the source in place of the node that was making the query. The “authoritative node” chosen was the most connected node in the network (in this case, the author). This metric was not significantly better than the control ($p > .11$). The second global metric calculated a t_{ij}' rating for the sink as the average rating given to it by direct neighbors. In figure 3, for example, the rating for the sink would always be 4 (the average of the two ratings assigned to it: 6 and 2). Like the other global metric, this one did not statistically outperform the control ($p > .36$), and had a very high standard deviation (1.45).

These results support the earlier hypotheses regarding the benefits of local metrics, and the effectiveness of this metric in particular. It also shows that we can expect the inferred value to be relatively close to the value a user would want. It is with these results confirming a high accuracy of our metric that we look toward applying it in the context of provenance.

4 Integrating Trust with Provenance

4.1 Trust Network Inferences to Rate Claims

In our prototype, we focus our attention on individual *claims*, with the key provenance information being the set of *claimants*. That is, we primarily deal with *knowledge provenance*. A claim is simply any RDF triple submitted to our website, whether by a Web form, via some aggregation mechanism such as an RSS feed, or by a Web service API. Triples are typically submitted in batches (i.e., in “snippets”) with user supplied metadata about the snippet inherited by each claim. As any triple can appear in any – and many – snippets, each claim can have multiple claimants. When a claimant is identifiable as a particular node in the trust network, we can attempt to determine a local trust rating for that claimant. If a user of the site has registered their trust network identifier with the site, then trust rating can be inferred with their node as the source. The webmaster for the website can also serve as a default source for the trust inference. The user might do this for the whole site to enjoy a personalized version of the site, or in interaction with particular pages. Given a particular trust rating for a claim, we customize the display behavior of the site. The simplest customization is to suppress the display of any claim from a claimant whose trust rating is below a certain, user configurable threshold.

The situation is slightly more complex if there are multiple claimants for a particular claim. There are a number of functions one could use to derive a trust rating for the *claim* based on the set of ratings for the claimants. However, the straightforward solution – take the maximum rating of the claimants – has a great deal of intuitive plausibility. Recall that, in our system, trust is a measure of confidence that the trusted person will in an ambiguous situation perform the action with a positive result. Since the kind of web sites we are building are community oriented portals, the general goal for the site is to be interesting, relevant, and useful to that community. Thus, the particular ambiguous situation, e.g., making a claim on the site, the potential positive result is well-presented, relevant, interesting, and useful information, as determined by the portal’s community standards. Since the trustworthiness of the claimant is *not* interpreted as evidence for or against the claim, there is no need to average or other balance divergent ratings.¹

4.2 Using Claim Ratings in Semantic Web Systems

The first and most obvious application of the ratings for claims is to filter the content of the website based on the value of the rating. Consider applying this technology in the context of some of the many “rumor” sites on the web. As one example,

¹ This accords with experience reported in [12]: “Our past experience with explanation systems for knowledge representation systems has shown that source metadata information for answers is the most important explanation feature for many users and, in many cases, it meets all or most of their explanation needs...they may only need source metadata information to trust answers.” In the describe system, trusting answers entails simply accepting them for the task at hand.

MacRumors² allows users to submit rumors about news and technology releases related to Apple Computer. The author of each rumor is tracked, and community members already have the ability to rate rumors as positive or negative. A website with that model would significantly benefit from a semantically aware system of trust and provenance. A network of ratings that reflect one person's opinion about the quality of posts made by another user, and following the system of generating ratings for statements based on their provenance creates the groundwork for allowing users to customize the site. Users can choose a minimum trust level of statements that appear on the site, and not only is the site personalized, but *optimized* for the user according to their preferences and social network connections. Although "rumor" sites provide an intuitive example because of the obvious variation in the credibility of statements, this technique clearly can be applied to any site where statements originate from a variety of sources.

Because trust networks are social networks, enhancing sociality of the network is critical. Community based and oriented portals likewise depend on the engagement of community members. For people to participate in a network or portal, the system has to be reasonably transparent to them, and the effort they put in to supply claims or trust ratings must have tangible and not-too-delayed rewards. Requiring trust ratings to reflect a rater's considered judgment of the epistemic reliability of community members is difficult and fraught with conceptual and pragmatic difficulties. Site personalization has a much simpler feedback loop, and the potential risks are both low and relatively non-threatening.

4.3 Filtering Inferences in Knowledge-Bases with Trust Values

Filtering the base claims of the system is useful and interesting, but base claims on Semantic Web sites form only part of the picture. Semantic Web portals tend to be oriented around RDFS and OWL ontologies, that is, logical theories of varying degrees of expressiveness. A Semantic Web site, therefore, is based on a knowledge base and the character of the web site is significantly influenced by the sorts of reasoning it supports. The ordering of filtering and inferring is important to consider. If the set of base claims are filtered first, and then inferences are made over the filtered set, there are two results. First, using the filtered base claims as the fact base for any inferences already filters the inferences. This allows users to conclude that any inferred statements should have at least the same trust rating as the minimum value in the filter, because all of the claims that allow the inference meet or exceed that minimum. However, this does *not* provide a mechanism for actually calculating a value for an inferred statement - it only sets a minimum bound. Using this filter-then-infer method also means that the set of inferred statements are limited - it is possible that many other statements could have been inferred from the unfiltered base.

If the order of inferring and filtering is reversed, so all of the inferences are made over the full set, and then the set of all statements - base and inferred - are filtered, the issue becomes more complex because it requires that some trust value be established for the inferred statements. The rating for an inferred statement should

² <http://macrumors.com/>

clearly be some combination of the statements that lead to the inference. However, a number of different statements could lead to an inference. Consider the following set of base claims in N3. Each statement is marked with a trust rating calculated from its claimants.

```
9   :Person      a owl:Class .

8   :SpouseOfStudent  a owl:Class;
8       rdfs:subClassOf :Person,
8       [
8           a owl:Restriction;
8           owl:allValuesFrom :Student;
8           owl:onProperty :marriedTo ],
8       [
8           a owl:Restriction;
8           owl:cardinality "1";
8           owl:onProperty :marriedTo ] .

7   :Student     a owl:Class;
7       rdfs:subClassOf :Person .

9   :University  a owl:Class .

6   :attendsUniversity  a owl:ObjectProperty;
6       rdfs:domain :Student;
6       rdfs:range :University .

10  :marriedTo   a owl:ObjectProperty;
10      owl:inverseOf :marriedTo .

10  :Daniel      a SpouseOfStudent;
9      <marriedTo> :Jennifer .

8   :Jennifer    a Person;
6       <attendsUniversity> :UMCP;
9       <marriedTo> :Daniel .
```

From this example, we can infer that `Jennifer` is a `Student`. What should be the rating for that inferred claim? There are several ways that it can be inferred. Because `Jennifer attendsUniversity UMCP` (known at level 6), and the domain of `attendsUniversity` is `Student` (rated at level 6), we can infer that `Jennifer` is a `Student`. This inference comes from two simple statements, rated at the same level, so it seems intuitive to rate the inference from these sources at a level 6, like the composite statements. There are other ways to infer that `Jennifer` is a `Student`, though, and they may have a higher rating than the 6 achieved with the first method. We also know that `Daniel`, a `SpouseOfStudent` (known at level 10), is `marriedTo Jennifer` (known at level 9). Since, for instances of the

`SpouseOfStudent` class, the object of `marriedTo` must be from the class `Student` (known at level 8), and that there must be exactly one spouse (because of the cardinality restriction known at level 8)– so `Jennifer` must be the only person that `Daniel` is `marriedTo` – we can infer that `Jennifer` is a `Student`. How to combine this series of statements into a rating for the inferred value is not as clear. There is also a third way of inferring the fact, stemming from the claim that `Jennifer` is `marriedTo` `Daniel` (rated at level 9) Because `marriedTo` is the inverse of itself (rated at level 10), we know that `Daniel` is `marriedTo` `Jennifer` (even if that were not explicitly stated). This leads to the second inference we made, allowing us to conclude that `Jennifer` is a `Student`.

This example illustrates several issues raised when considering how to rate inferred statements. First, for each set of statements that leads to an inference, there must be a way to combine the ratings of the composite statements to come up with a rating for the inferred statement. Even if we took the simple route of just using the minimum rating from the set of composite statements as the rating for the inferred statement, there are still more problems. If a statement is inferred from several sets of statements, there are now several ratings for that inferred statement. How to choose a final value for the inferred statement is not clear – it could be the maximum value from all of the possible values assigned or some composite. On top of that, the primary issue illustrated by the above example is that the number of ways a statement can be inferred can grow very quickly. To consider every possible combination of claims that lead to an inference could become computationally difficult. Because inferences are such a fundamental issue on the Semantic Web, the question of establishing trust values for inferred statements space will be the focus of future work in this space.

5 Conclusions and Future Work

In this paper, we have presented a accurate system for inferring reputation in semantic web based trust networks, and using those values to create ratings for statements made by individuals within the network. By combining trust values with provenance information, we show how users can filter knowledge bases based on a minimum trustworthiness rating.

Refining the trust metric is one point of future work. Though this analysis has shown that our simple metric is relatively accurate, considering additional structural features of the network such as path length, number of paths, and the use of intermediate nodes, may lead to more accurate metrics. Understanding which features of a trust inference algorithm should be incorporated for the most accurate metric will be an important step as this work progresses.

While we present the idea of using this system of ratings over inferences, a major future step will be to address the issue of ratings for statements derived through inferences. Section 4 presented a detailed argument for why creating these ratings is difficult. Further analysis of this issue will be critical in extending this work into a broader application on the Semantic Web.

References

1. da Silva, Paulo Pinheiro, Deborah L. McGuinness and Rob McCool. Knowledge Provenance Infrastructure. IEEE Data Engineering Bulletin Vol.26 No.4, pages 26-32, December 2003.
2. Dudek, C. (2003). "Visual Appeal and the Formation of Trust in E-commerce Web Sites." Unpublished Masters Thesis, Carleton University, Ottawa, Canada.
3. Dumbill, Ed, "XML Watch: Finding friends with XML and RDF." IBM Developer Works, <http://www-106.ibm.com/developerworks/xml/library/x-foaf.html>, June 2002. [accessed 2004]
4. Gil, Yolanda and Varun Ratnakar, "Trusting Information Sources One Citizen at a Time," *Proceedings of the First International Semantic Web Conference (ISWC)*, Sardinia, Italy, June 2002.
5. Golbeck, Jennifer, Bijan Parsia, James Hendler, "Trust Networks on the Semantic Web," *Proceedings of Cooperative Intelligent Agents 2003*, Helsinki, Finland, August 27-29.
6. Kamvar, Sepandar D. Mario T. Schlosser, Hector Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks", *Proceedings of the 12th International World Wide Web Conference*, Budapest, Hungary, May 20-24, 2003.
7. Kleinberg, J, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, 1999.
8. Kumar, Ravi, Prabhakar Raghavan, Sridhar Rajagopalan, D. Sivakumar, Andrew Tomkins, and Eli Upfal. "The web as a graph". *Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, May 15-17, 2000.
9. Labalme, Fen, Kevin Burton, "Enhancing the Internet with Reputations: An Openprivacy Whitepaper," <http://www.openprivacy.org/papers/200103-white.html>, March 2001. [accessed April 2004]
10. Levien, Raph and Alexander Aiken. "Attack resistant trust metrics for public key certification." *7th USENIX Security Symposium*, San Antonio, Texas, January 1998.
11. Marsh, S. "Formalising Trust as a Computational Concept." PhD thesis, Department of Mathematics and Computer Science, University of Stirling, 1994
12. McGuinness Deborah L., and Paulo Pinheiro da Silva. Trusting Answers from Web Applications. *To appear in Mark T. Maybury, editor, New Directions in Question Answering*. AAAI/MIT Press.
13. Milgram, Stanley. "The Small World Problem." *Psychology Today*, May 1967: 60 – 67.
14. Page, L., Brin, S., Motwani, R., & Winograd, T. "The PageRank citation ranking: Bringing order to the web." Technical Report 1998, Stanford University, Stanford, CA.
15. RDFWeb: FOAF: 'The Friend of a Friend Vocabulary', <http://xmlns.com/foaf/0.1/>
16. RDFWeb: Co-depiction Photo Meta Data: <http://rdfweb.org/2002/01/photo/>
17. Richardson, Matthew, Rakesh Agrawal, Pedro Domingos. "Trust Management for the Semantic Web," *Proceedings of the Second International Semantic Web Conference*, Sanibel Island, Florida, 2003.
18. Stevens, R., A. Robinson, and C.A. Goble "myGrid: Personalised Bioinformatics on the Information Grid" in *proceedings of 11th International Conference on Intelligent Systems in Molecular Biology*, 29th June–3rd July 2003, Brisbane, Australia, published Bioinformatics Vol. 19 Suppl. 1 2003, i302-i304
19. The Trust Ontology: <http://trust.mindswap.org/ont/trust.owl>